Journal of Computer and System Sciences 144 (2024) 103548

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

journal homepage: www.elsevier.com/locate/jcss

On computing large temporal (unilateral) connected components ☆

Isnard Lopes Costa [a,*], Raul Lopes [b,c,*], Andrea Marino [d,*], Ana Silva [a,d,*]

[a] Departamento de Matemática, Universidade Federal do Ceará, Fortaleza, CE, Brazil
[b] Université Paris-Dauphine, PSL University, CNRS UMR7243, LAMSADE, Paris, France
[c] DIENS, École normale supérieure de Paris, CNRS, Paris, France
[d] Dipartimento di Statistica, Informatica, Applicazioni, Università degli Studi di Firenze, Firenze, Italy

ARTICLE INFO

Article history:
Received 31 July 2023
Received in revised form 22 March 2024
Accepted 6 May 2024
Available online 22 May 2024

Keywords:
Temporal graphs
Components
Graph algorithms

ABSTRACT

A temporal (directed) graph is one where edges are available only at specific times during its lifetime, τ. Paths in these graphs are sequences of adjacent edges whose appearing times are either strictly increasing or non-strictly increasing. Classical concepts of connected and unilateral components can be naturally extended to temporal graphs. We address fundamental questions in temporal graphs. (i) What is the complexity of deciding the existence of a component of size k, parameterized by τ, k, and $k+\tau$? The answer depends on the component definition and whether the graph is directed or undirected. (ii) What is the minimum running time to check if a subset of vertices is pairwise reachable? A quadratic-time algorithm exists, but a faster time is unlikely unless SETH fails. (iii) Can we verify if a subset of vertices forms a component in polynomial time? This is NP-complete depending on the temporal component definition.

1. Introduction

A *(directed) temporal graph* (G, λ) *with lifetime* τ consists of a (directed) graph G together with a *time-function* $\lambda : E(G) \to 2^{[\tau]}$ which tells when each edge $e \in E(G)$ is available along the discrete time interval $[\tau]$. Given $i \in [\tau]$, the *snapshot* G_i refers to the subgraph of G containing exactly the edges available in time i. Temporal graphs, also appearing in the literature under different names [1–3], have attracted a lot of attention in the past decade, as many works have extended classical notions of graph theory to temporal graphs (we refer the reader to the surveys [3,4] and the seminal paper [5]).

A crucial characteristic of temporal graphs is that a u, v-walk/path in G is valid only if it traverses a sequence of adjacent edges $e_1, \dots, e_k$ at non-decreasing times $t_1 \le \dots \le t_k$, respectively, with $t_i \in \lambda(e_i)$ for every $i \in [k]$. Similarly, one can consider strictly increasing sequences, i.e. with $t_1 < \dots < t_k$. The former model is referred to as *non-strict* model, while the latter as *strict*. In both settings, we call such sequence a *temporal* u, v*-walk/path*, and we say that u *reaches* v. For instance, in Fig. 1,

☆ Funded by: FUNCAP MLC-0191-00056.01.00/22 and PNE-0112-00061.01.00/16, CNPq 303803/2020-7, Italian PNRR CN4 Centro Nazionale per la Mobilità Sostenibile, NextGeneration EU-CUP, B13C22001000001, by MUR of Italy, under PRIN Project n. 2022ME9Z78-NextGRAAL: Next-generation algorithms for constrained GRAph visuALization, by PRIN PNRR Project n. P2022NZPJA-DLT-FRUIT: A user centered framework for facilitating DLTs FRUITion, (partially) supported by the group Casino/ENS Chair on Algorithmics and Machine Learning and French National Research Agency under JCJC program (ASSK: ANR-18-CE40-0025-01). Thanks to Giulia Punzi and Mamadou Kanté for interesting discussions.

* Corresponding authors.
E-mail addresses: isnard.lopes@alu.ufc.br (I.L. Costa), raul.wayne@gmail.com (R. Lopes), andrea.marino@unifi.it (A. Marino), anasilva@mat.ufc.br (A. Silva).

https://doi.org/10.1016/j.jcss.2024.103548

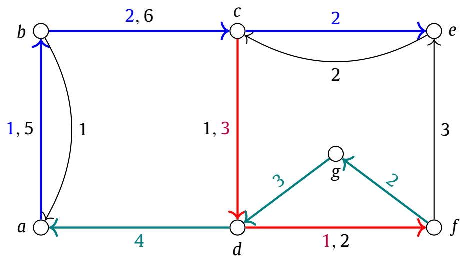

Fig. 1. A temporal graph, where on each edge e we depict $\lambda(e)$. Some of its components according to the non-strict model are reported below. $A' = \{a, b\}$ is a closed connected set, as a and b reach each other without using external vertices. $A = \{a, b, c, d\}$ is a maximal closed connected set, i.e. a closed TCC. $B = \{a, b, c, d, e\}$ is a closed TUCC but not a closed TCC as, using only vertices in B, a, b, c, d reach each other, e reaches all the vertices in B and vice versa, except for d, which does not reach e. B is also a TCC, as d can reach e using the external vertex f. $C = \{a, b, c, d, e, f\}$ is a TUCC as B forms a closed TUCC, f is able to reach every other vertex directly or via the external vertex g. However, C is not a TCC as a, b, c, e cannot reach f. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

both blue and green paths are valid in the non-strict model, but only the green one is valid in the strict model, as the blue one traverses two edges with label 2. The red path is not valid in either model.

The non-strict model is more appropriate in situations where the time granularity is relatively big. This is the case in a disease-spreading scenario [6], where the spreading speed might be unclear or in "time-varying graphs", as in [7], where a single snapshot corresponds to all the edges available in a time interval, e.g. the set of all the streets available in a day. As for the strict model, it can represent the connections of the public transportation network of a city which are available only at precise scheduled times. All in all, there is a rich literature on both models (see [8–11,6]), and this is why we explore both settings.

Connected Sets and Components. Given a temporal graph $\mathcal{G} = (G, \lambda)$, we say that $X \subseteq V(G)$ is a *temporal connected set* if u reaches v *and* v reaches u, for every $u, v \in X$. Extending the classical notion of connected components in static graphs, in [12] the authors define a *temporal connected component* (TCC for short) as a maximal connected set of $\mathcal{G}$. Such constraint can be strengthened to the existence of such paths using only vertices of X. Formally, X is a *closed temporal connected component* (closed TCC for short) if, for every $u, v \in X$, we have that u reaches v *and* v reaches u through temporal paths whose vertex sets are contained in X. See Fig. 1 for an example of a TCC and of a closed TCC.

Unilateral Connected Components. In the same fashion, also the concept of *unilateral connected component* can be extended to temporal graphs. In static graph theory, they are a well-studied relaxation of connected components which asks for a path from u to v or vice versa, for every pair u, v in the component [13,14]. More formally, in a directed graph G, we say that $X \subseteq V(G)$ is a *unilateral connected set* if u reaches v or v reaches u, for every $u, v \in X$. X is a *unilateral connected component* if it is maximal. In this paper, we introduce the definition of a *(closed) unilateral temporal connected set/component*, which can be seen as the immediate translation of unilateral connected component to the temporal context. Formally, $X \subseteq V(G)$ is a *temporal unilateral connected set* if u reaches v *or* v reaches u, for every $u, v \in X$, and it is a *closed unilateral connected set* if this holds using paths contained in X. Finally, a *(closed) temporal unilateral connected component* ((closed) TUCC for short) is a maximal (closed) temporal unilateral connected set. See again Fig. 1 for an example.

Problems In this paper, we deal with four different definitions of temporal connected components, depending on whether they are unilateral or not, and whether they are closed or not. In what follows, we pose three questions, and we comment on partial knowledge about each of them. Later on, we discuss our results, which close almost all the gaps found in the literature. For each of the types of components described above, we consider the natural problem of deciding if a given (directed) temporal graph contains a component of size at least k. For (closed) TCCs and (closed) TUCCs, we define the (CLOSED) TCC and (CLOSED) TUCC problems. Unless stated otherwise, the statements refer to undirected temporal graphs. When applied to directed temporal graphs, this is explicitly stated. We start by asking the following.

Question 1 *(Parameterized complexity). What is the complexity of deciding the existence of temporal components of size at least k parameterized by* (i) τ*, i.e. the lifetime,* (ii) k*, and* (iii) $k + \tau$*?*

In order to answer Question 1 for the strict model, there is a very simple parameterized reduction from k-clique, known to be W [1]-hard when parameterized by k [15], to decide the existence of connected components (both closed or not and both unilateral or not) of size at least k in undirected temporal graphs. This reduction has appeared in [16]. Given an undirected graph G, we can simply consider the temporal graph $\mathcal{G} = (G, \lambda)$ where $\lambda(uv) = \{1\}$ for all $uv \in E(G)$ (i.e., $\mathcal{G}$ is equal to G itself). As u temporally reaches v if and only if $uv \in E(G)$, one can see that all those problems are now equivalent to decide the existence of a k-clique in G. Observe that we get W[1]-hardness when parameterized by k or $k + \tau$, and para-NP-completeness when parameterized by τ, both in the undirected and the directed case.[1] However, this reduction

[1] In the directed case, it suffices to replace each edge of the input graph with two opposite directed edges between the same endpoints.

Table 1
A summary of our results for the parameterized complexity of computing components of size at least k of a temporal graph $\mathcal{G}$ having lifetime τ in the *non-strict* model. "W[1]-h" stands for W[1]-hardness and "p-NP" stands for para-NP-completeness. For the strict model the entries are W[1]-h in the third and fourth columns and p-NP in the second one already for $\tau = 1$, both for the directed and the undirected case.

	Par. τ	Par. k	Par. $k + \tau$
TCC	p-NP $\tau \geq 2$ Undir. (Th. 1) and Dir. (Th. 3)	W[1]-h Dir. $\tau \geq 2$ (Th. 3) and Undir. (Th. 2)	W[1]-h Dir. (Th. 3)
TUCC		W[1]-h Dir. $\tau \geq 2$ (Th. 3) FPT Undir. (Th. 5)	FPT Undir. (Th. 5)
closed TCC		W[1]-h Dir. $\tau \geq 3$ (Th. 4)	W[1]-h Dir. (Th. 4) FPT Undir. (Th. 5)
closed TUCC		W[1]-h Dir. $\tau \geq 3$ (Th. 4) FPT Undir. (Th. 5)	

does not work in the case of the *non-strict* model, leaving Question 1 open. Indeed the reductions in [12] and in [17] for (closed) TCCs, which work indistinctly for both the strict or the non-strict models, are not parameterized reductions. We also observe that the aforementioned reductions work on the non-strict model only for $\tau \geq 4$.

Another question of interest is the following. Letting n be the number of vertices in $\mathcal{G}$ and M be the number of *temporal edges*,[2] it is known that, in order to verify whether $X \subseteq V(G)$ is a connected set in $\mathcal{G}$, we can simply apply $O(n)$ single source "best" path computations (see e.g. [18,19]), resulting in a time complexity of $O(n \cdot M)$. This is $O(M^2)$ if $\mathcal{G}$ has no isolated vertices, a natural assumption when dealing with connectivity problems. As in static graphs testing connectivity can be done in linear time [20], we ask whether the described algorithm can be improved.

Question 2 *(Lower bound on checking connectivity). Given a temporal graph $\mathcal{G}$ and a subset $X \subseteq V(\mathcal{G})$, what is the minimum running time required to check whether X is a (unilateral) connected set?*

Finally we focus on one last question.

Question 3 *(Checking maximality). Given a temporal graph $\mathcal{G}$ and a subset $X \subseteq V(\mathcal{G})$, is it possible to verify, in polynomial time, whether X is a component, i.e. a maximal (closed) (unilateral) connected set?*

For Question 3, we first observe that the property of being a temporal (unilateral) connected set is hereditary (forming an independence system [21], see [22] for a survey about set systems), meaning that every subset of a (unilateral) connected set is still a (unilateral) connected set. For instance, in Fig. 1, every subset of the connected set $B = \{a, b, c, d, e\}$ is a connected set. Also, checking whether $X' \subseteq V(G)$ is a temporal (unilateral) connected set can be done in time $O(n \cdot M)$, as discussed above. We can then check whether X is a maximal such set in time $O(n^2 \cdot M)$: it suffices to test, for every $v \in V(G) \setminus X$, whether by adding v to X we still get a temporal (unilateral) connected set. On the other hand, *closed* connected (unilateral) sets are not hereditary, because by removing vertices from the set we could destroy the paths between other members of the set. This is the case for the closed connected set $A = \{a, b, c, d\}$ in Fig. 1, since by removing d there are no temporal paths from c to a nor b using only vertices in the remainder of the set. This implies that the same approach as before does not work, i.e., we cannot check whether X is maximal by adding to X a *single* vertex at a time, then checking for connectivity. For instance, the closed connected set $A' = \{a, b\}$ in Fig. 1 cannot be grown into the closed connected set A by adding one vertex at a time, since both $A' \cup \{c\}$ and $A' \cup \{d\}$ are not closed connected sets. Hence, the answer to Question 3 for closed sets does not seem easy, and until now was still open.

We remark the important practical consequences of the latter question. Indeed, in practice, when trying to find structures of maximum size, a usual viable strategy is modifying *backtracking listing algorithms* for efficient generating maximal structures (eventually with pruning strategies) and choosing the largest structures found [23,24]. Such algorithms typically solve the so-called *extension problem*, that is generating all (or some of) the maximal solutions enlarging a partial one [25,26,22,27]. Question 3 implicitly asks whether efficient generation of closed TCCs or closed TUCCs is likely to exist or not.

Our results Our results concerning Question 1 are reported in Table 1 for the non-strict model, since for the strict model all the entries would be W [1]-hard or para-NP-complete already for $\tau = 1$, as we argued before. In the non-strict model, we observe instead that the situation is much more granulated. If $\tau = 1$, then all the problems become the corresponding traditional ones in static graphs, which are all solvable in polynomial time (see Paragraph "Related works"). As for bigger values of τ, the complexity depends on the definition of component being considered, and whether the temporal graph is

[2] $M = \sum_{e \in E(\mathcal{G})} |\lambda(e)|$.

Table 2
Our results for Question 2 and Question 3, holding for both the *strict* and the *non-strict* models. Recall that a component is a (inclusion-wise) maximal connected set. *The $O(\cdot)$ result can be obtained by applying any "single source best path" algorithm (e.g. the ones presented in [18,19]), as explained in the paragraphs preceding Question 2 and succeeding Question 3. M (resp. n) denotes the number of temporal edges (resp. nodes) in $\mathcal{G}$.

	Check whether $X \subseteq V$ is a connected set	Check whether $X \subseteq V$ is a component
TCC	$\Theta(M^2)$ (Th. 6)	$O(n^2 \cdot M)$*
TUCC		
CLOSED TCC		NP-c (Th. 7)
CLOSED TUCC		

directed or not. Table 1 considers $\tau > 1$, reporting on negative results, "$\tau \geq x$" for some x meaning that the negative result starts to hold for temporal graphs of lifetime at least x.

The second column of Table 1 addresses Question 1*(i)*, i.e., parameterization by τ. We prove that, for all the definitions of components being considered, the related problem becomes immediately para-NP-complete as soon as τ increases from 1 to 2 for both the undirected and directed cases; this is done with Theorems 1 and 3. Theorem 1 improves upon the reduction of [12], which holds only for $\tau \geq 4$. We remark that a similar result has also appeared online in a very recent manuscript [28], which also have independently proved NP-completeness for $\tau = 2$ for (closed) TCC. The relation of this work with our is discussed in the related work section.

Question 1*(ii)* (parameterization by k) is addressed in the third column of Table 1. Considering first directed temporal graphs, we prove that all the problems are W[1]-hard. In particular, deciding the existence of a TCC or TUCC of size at least k is W[1]-hard already for $\tau \geq 2$ (Theorem 3). As for the existence of closed components, W[1]-hardness also holds as long as $\tau \geq 3$ (Theorem 4). Observe that, since τ is constant in both results, these also imply the W[1]-hardness results presented in the last column, thus answering also Question 1*(iii)* (parameterization by $k + \tau$) for directed graphs. On the other hand, if the temporal graph is undirected, then the situation is even more granulated. Deciding the existence of a TCC of size at least k remains W[1]-hard, but only if τ is unbounded. This is complemented by the answer to Question 1*(iii)*, presented in the last column of Table 1: TCC and (even) closed TCC are FPT on undirected temporal graphs when parameterized by $k + \tau$ (Theorem 5). We also give FPT algorithms when parameterized by k for unilateral components, namely TUCC and (closed) TUCC. Observe how this differs from TCC, whose corresponding problem is W[1]-hard, meaning that unilateral and traditional components behave very differently when parameterized by k.

In summary, Table 1 answers Question 1 for almost all the definitions of components, both for directed and undirected temporal graphs. We leave open only the following three problems. Given an undirected temporal graph, deciding the existence of a closed TCC of size at least k when parameterized by k. And, given a directed temporal graph with lifetime 2, deciding the existence of closed TCC (closed TUCC) of size at least k when parameterized by k.

Concerning Questions 2 and 3, our results are summarized in Table 2. All these results hold both for the strict and the non-strict models. For Question 2, we prove that the trivial $O(M^2)$ algorithm to test whether S is a (closed) (unilateral) connected set is best possible, unless the Strong Exponential Time Hypothesis (SETH) fails [29]. For Question 3, in the case of TCC and TUCC, we have already seen that checking whether a set $X \subseteq V$ is a component can be done in $O(n^2 \cdot M)$. Interestingly, for closed TCC and closed TUCC, we answer negatively (unless P =NP) to Question 3.

Related work The notions of *temporal connected component* and *closed temporal connected component* have been defined in [12], where the authors proved that, given a temporal graph $\mathcal{G}$ and a positive integer k, deciding whether $\mathcal{G}$ has a (closed) TCC of size at least k is NP-complete even if $\mathcal{G}$ has lifetime 4. Their construction works indistinctly for both the strict or the non-strict setting. Such problems continue to be NP-complete even if the underlying graph has geometric properties, namely even if it is a unit disc graph, as shown in [30], and even if each edge appears exactly once and each snapshot forms a matching (these are called *happy graphs*), as shown in [17] both for the strict or the non-strict setting, leaving open the case when $\tau = 2$ or 3. In [28, Theorem 34 and 37], the authors close also the cases $\tau = 2$ and 3 as we do independently in Theorem 1. However, all these reductions are not parametrized and do not imply W [1]-hardness wrt k as the ones we show in this paper (third column of Table 1). In [28], they also provide an algorithm for TCC running in time $O(nm \log \tau + \min\{n^k k^2, 2^{0.25n}\})$ and an algorithm for closed TCC running in time $O(2^n n(m \log \tau + n \log n))$, where $n = |V(G)|$ and $m = |E(G)|$, both on undirected graphs. These algorithms are either XP when parameterized by k or exponential on n, while the algorithms we provide in this paper are FPT in $k + \tau$ (fourth column of Table 1).

In [7], the authors show that the problem of computing TCCs is a particular case of finding cliques in the so-called *affine graph*. This does not imply that the problem is NP-complete as claimed, since in order to prove hardness one needs to do a reduction on the opposite direction, i.e., a reduction from a hard problem to TCC instead.

There are many other papers about temporal connected components in the literature. In [16], by constructing on Moon-Moser [31] graphs, they provide examples of temporal graphs on n vertices that have $2^{\Theta(\sqrt{n})}$ temporal connected components, both for the strict and the non-strict models. New improved bounds for the number of such components have been given in [28]. Further related works include recent papers on giant components and connectivity in randomized temporal graphs [32,33], as well as papers on other notions of temporal components as for instance temporal in/out-component [34], Δ-component [35], persistent components [36], and weakly connected components [34]. *Weakly connected components* are only defined if the temporal graph is directed; as its analogue static version, given a directed temporal graph $\mathcal{G}$, it is said that u and v are weakly temporally connected in $\mathcal{G}$ is there is temporal u, v-path in the underlying undirected temporal graph related to $\mathcal{G}$. In this paper, we implicitly consider also weakly temporal connected components as studying how to compute them in directed temporal graphs is the same as studying how to compute (closed) TCCs in undirected temporal graphs. As for the *temporal out-components* (resp. *in-components*) defined in [34], these are the sets of vertices which can be reached from (resp.reaching) a vertex in a time-varying graph. Components similar to these have also been considered in [28]. The Δ*-components* in [35] can intuitively be seen as subsets of vertices that are connected by paths that last at most Δ timesteps throughout the lifetime. Finally, *persistent connected components* [36] are quadruplets specifying the set of vertices involved, its size, and the time window where such vertices are connected. In the same paper, they consider networks with continuous varying-time with interval assigned to the edges. An interval temporal network is connected during a period of time $[x, y]$ if it is connected for all timesteps $t \in [x, y]$. In [37], they provide a polynomial-time algorithm that answers whether the temporal network is connected during a time period. If the network is not connected throughout the given time period, then they give a polynomial-time algorithm that return large components of the networks that remain connected and remains large during $[x, y]$. They also examine a case of interval temporal graphs networks on trees, where the lifetimes of links are not controlled; and they show that one can with high probability maintain the connectivity of the network for a long time period.

Finally, we remark that there are many results in the literature concerning unilateral components in static graphs, also with applications to community detection [38]. Even though the number of unilateral components in a graph is exponential [13], deciding whether there is one of size at least k is polynomial. In [13, Theorem 3], they prove that this corresponds to decide whether a DAG with weights on its vertices has a path of weight at least k, which in turn can be done in polynomial time by slightly modifying the algorithm for longest paths [39]. In [40,13], they propose an algorithm for finding all the unilateral components of a digraph on n vertices and m edges that runs in time $O(m + nc)$ and uses $O(m + n)$ space, where c denotes the number of unilateral components. In [38], the authors propose a method to find communities in directed networks based on strong components and unilateral components. In [41], a characterization of unilateral connected graphs is presented. In [42], they give a polynomial-time algorithm to recognize whether the non-directed edges of a mixed graph can be oriented in a way as to obtain a unilateral connected directed graph. In [43], they relate unilateral connected orientations and traceability of graphs (see also [41], Theorem 5.11.5). There were no results about unilateral components in temporal graphs until now.

Structure of the paper In Section 2 we introduce basic notations and definitions on static and temporal graphs, as well as some preliminary results. In Section 3 we prove the results mentioned in Table 1. Section 4 is devoted to the lower bounds considered in Question 2. In Section 5 we prove the results mentioned in the second column of Table 2, showing that it is hard to decide if a given set of vertices of a temporal graph, directed or not, forms a (closed) TCC or TUCC. Finally, our concluding remarks and open questions are contained in Section 6.

2. Preliminaries

For basic graph theory concepts and notation, we refer to [44], and to [45,46] for basic background on parameterized complexity.

2.1. Static graphs

Given a graph $G = (V, E)$, directed or not, and a set $X \subseteq V(G)$ we write $G[X]$ for the subgraph of G induced by X. If e is an edge of a directed or undirected graph with *endpoints* u and v, we may refer to e as uv and say that e is *incident* to u and v. If e is an edge of a directed graph[3] from u to v of a directed graph, we say that e is from u to v, and is *oriented* from u to v. The *degree* $d_G(v)$ of a vertex v of a (directed) graph G is the number of edges of G incident to v. We denote by $\Delta(G)$ the maximum degree of the vertices of G. The *neighborhood* $N_G(v)$ of v is the set $\{u \in V(G) \mid uv \in E(G)\}$, while the *closed neighborhood* of v in G is the set $N_G[v] = N_G(v) \cup \{v\}$. We omit G if it is clear from the context. If $X, Y \subseteq V(G)$ are subsets of vertices and G is clear from the context, we denote by $N_Y(X)$ the set $\{y \in Y \setminus X \mid y \in N_G(x)$ for some $x \in X\}$. If D is a digraph, the *in-neighborhood* $N^-_D(v)$ of v is the set $\{u \in V(D) \mid uv \in E(D)\}$, and the *out-neighborhood* $N^+_D(v)$ is the set $\{u \in V(D) \mid vu \in E(D)\}$.

An undirected graph G is said to be *simple* if there is at most one edge between every pair of vertices of G (i.e., there are no *parallel* edges). We say that G is *bipartite* if there is a partition of $V(G)$ into two non-empty sets X and Y such that every edge of G has one endpoint in X and the other endpoint in Y. A *matching* of G is a set of edges $M \subseteq E(G)$ such that no two edges of M share an endpoint; i.e., they all are pairwise *independent*. This definition also applies to oriented edges

[3] We refer to arcs of directed graphs as edges (following the notation in [44]).

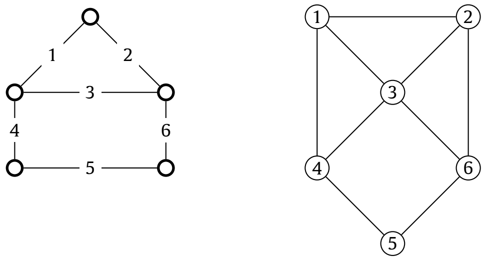

Fig. 2. To the left, a graph G, to the right its line graph, L.

in directed graphs. A *clique* in an graph H is a subset $C \subseteq V(H)$ such that all vertices of C are pairwise adjacent. If H is directed, we say that C is a clique (resp. *full clique*) if one of (resp.both) the two possible edges exist between u and v, for every pair $u, v \in C$, $u \neq v$. A *biclique* in a bipartite graph H is a disjoint pair of sets $A, B \subseteq V(H)$ such that there is an edge from every $a \in A$ to every $b \in B$. A graph is $2K_2$*-free* if it does not contain a pair of edges uv and xy such that $G[\{u, v, x, y\}]$ contains exactly the two edges (i.e., is isomorphic to a $2K_2$).

An *orientation* of an undirected graph G is a digraph D obtained from G by choosing an orientation for each edge $e \in E(G)$. The undirected graph G formed by ignoring the orientation of the edges of a digraph D is the *underlying graph* of D. The *line graph* of an undirected graph G is the graph L with vertex set $E(G)$ where two vertices $e, f \in V(L)$ associated with edges of G are linked by an edge if and only if e and f share an endpoint in G. See Fig. 2 for an example. For a undirected graph G and $u \in V(G)$, let $\delta(u)$ denote the set $\{e \in E(G) \mid u \text{ is incident to } e\}$. For a set $S \subseteq E(G)$, the *edge-induced subgraph* $G[S]$ is a graph whose edge set is S and vertex set consists of all endpoints of the edges in S.

Given a (directed) graph G and vertices $v_0, v_q \in V(G)$, a v_0, v_q*-walk* in G is a sequence of vertices and edges, $P = (v_0, e_1, v_1, \cdots, e_q, v_q)$, such that e_i has endpoints $v_{i-1}v_i$, for each $i \in [q]$. Additionally, if no vertices of G are repeated in P, then we say that P is a v_0, v_q*-path*. A graph G is *connected* if, for every pair of vertices $u, v \in V(D)$, there is a walk between u and v in G. A *connected component of* G is a maximal connected subgraph of G. A directed graph D is *strongly connected* if, for every pair of vertices $u, v \in V(D)$, there is a u, v-walk and a v, u-walk in D. We say that D is *weakly connected* if the underlying graph of D is connected. A *strong component* of D is a maximal induced subgraph of D that is strongly connected, and a *weak component* of D is a maximal induced subgraph of D that is weakly connected. For simplicity, we call a strong connected component of a directed graph simply "connected component".

2.2. Parameterized complexity

A *parameterized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$. For an instance $I = (x, k) \in \Sigma^* \times \mathbb{N}$, k is called the *parameter*. A parameterized problem L is *fixed-parameter tractable* (FPT) if there exists an algorithm $\mathcal{A}$, a computable function f, and a constant c such that given an instance $I = (x, k)$, $\mathcal{A}$ (called an FPT *algorithm*) correctly decides whether $I \in L$ in time bounded by $O(f(k) \cdot |I|^c)$.

A parameterized problem L is in XP if there exists an algorithm $\mathcal{A}$ and two computable functions f and g such that given an instance $I = (x, k)$, $\mathcal{A}$ (called an XP *algorithm*) correctly decides whether $I \in L$ in time bounded by $O(f(k) \cdot |I|^{g(k)})$. For instance, the Clique problem, i.e. deciding whether there exists a clique of size at least k, parameterized by k is in XP.

Within parameterized problems, the class W[1] may be seen as the parameterized equivalent to the class NP of classical decision problems. Without entering into details (see [46,45] for the formal definitions), a parameterized problem being W[1]-*hard* can be seen as a strong evidence that this problem is *not* FPT. Clique parameterized by the size of the solution is the canonical example of a W[1]-hard problem.

Parameterized reductions are used to transfer fixed-parameter tractability or hardness between parameterized problems. Namely, a parameterized reduction is an algorithm that, given an instance (x, k) of a parameterized problem L, runs in time $f(k) \cdot |x|^{O(1)}$ and outputs an instance (x', k') of a parameterized problem L' such that $k' \leq g(k)$ for some computable function g and (x, k) is positive if and only if (x', k') is positive. For example, if L is W[1]-hard and there is a parameterized reduction from L to L', then L' is also W[1]-hard and thus unlikely to admit an FPT algorithm.

2.3. Temporal graphs, paths and components

Recall that a *(directed) temporal graph with lifetime* τ is a pair (G, λ) where G is a (directed) graph and λ is a function from $E(G)$ to $2^{[\tau]}$, also called *time-function*. Additionally, given $i \in [\tau]$, the *snapshot* G_i is the subgraph $(V(G), E_i)$, where $E_i = \{uv \in E(G) \mid i \in \lambda(uv)\}$.

Given a temporal (directed) graph $\mathcal{G} = (G, \lambda)$ and vertices $v_0, v_q \in V(G)$, a *temporal* v_0, v_q*-walk* in $\mathcal{G}$ is defined as a sequence of vertices and temporal edges, $(v_0, \alpha_1, v_1, \cdots, \alpha_q, v_q)$ such that, for each $i \in [q]$, α_i has endpoints $v_{i-1}v_i$ and is active in a timestep t_i which is at most equal to the timestep where α_{i+1} is active. Sometimes we abuse notation and write $P = (v_0, t_1, v_1, \ldots, t_q, v_q)$ instead, where t_i is equal to the timestep where α_i is active, for every $i \in [q]$. We then say that P *starts in time* t_1 and *finishes in time* t_q. Given $i, j \in \{0, \ldots, q\}$, we denote by $v_i P v_j$ the v_i, v_j-walk $(v_i, t_{i+1}, v_{i+1}, \ldots, t_j, v_j)$.

Additionally, if no vertices of G are repeated in P, then we say that P is a *temporal* v_0, v_q*-path*. It is important to mention that distinctions between paths and walks are important for some problems, but since it is not the case in this work, the reader should not worry about interchangeable uses of "walks" and "paths" along the text. Given two vertices $u, v \in V(G)$, we say that u *reaches* v in $\mathcal{G}$ if there exists a temporal u, v-walk in $\mathcal{G}$.

Given a temporal (directed) graph $\mathcal{G} = (G, \lambda)$ and a subset $S \subseteq V(G)$, we say that S is *temporal connected* (in $\mathcal{G}$) if there is a temporal u, v-path in $\mathcal{G}$ for every ordered pair $(u, v) \in S \times S$. In [12], the authors define a *temporal connected component* (TCC for short) as a maximal subset $S \subseteq V(G)$ such that S is temporal connected. Similarly, a *closed temporal connected component* (closed TCC for short) was defined as a maximal subset $S \subseteq V(G)$ for which, for every ordered pair $(u, v) \in S \times S$, there is a temporal u, v-path in $\mathcal{G}$ using only vertices of S. In other words, a closed TCC is a maximal subset $S \subseteq V(G)$ such that $\mathcal{G}[S]$ (the temporal subgraph induced by S) is temporal connected. We say that S is a *temporal unilateral connected set* if for every pair $u, v \in S$ there is a temporal u, v-path or a temporal v, u-path in $\mathcal{G}$. If all such paths use only vertices in S then we say that S is a *closed temporal unilateral connected set*. If S is maximal such set, then we say that S is a *temporal unilateral connected component* (TUCC) in the first case, and a *closed temporal unilateral connected component* (closed TUCC) in the second case.

The *reachability digraph* $\mathrm{R}(\mathcal{G})$ associated to $\mathcal{G}$ is a directed graph with the same vertex set as $\mathcal{G}$, and such that uv is an edge in $\mathrm{R}(\mathcal{G})$ if and only if u reaches v in $\mathcal{G}$, $u \neq v$. This is a slight generalization of the *affine* graph introduced in [7]. There, since they are interested only in the TCC variants, they consider pairs that are mutually reachable from each other, ignoring the edges uv of $\mathrm{R}(\mathcal{G})$ that are not symmetric (i.e., for which vu is not present).

The following result is an immediate consequence of the definition of reachability graph $\mathrm{R}(\mathcal{G})$.

Lemma 1. *Given a temporal (directed) graph $\mathcal{G} = (G, \lambda)$, then the following hold:*

1. *C is a TCC in $\mathcal{G}$ if and only if C is a maximal full clique in $\mathrm{R}(\mathcal{G})$;*
2. *C is a TUCC in $\mathcal{G}$ if and only if C is a maximal clique in $\mathrm{R}(\mathcal{G})$;*
3. *If C is a closed TCC in $\mathcal{G}$, then C is a full clique in $\mathrm{R}(\mathcal{G})$; and*
4. *If C is a closed TUCC in $\mathcal{G}$, then C is a clique in $\mathrm{R}(\mathcal{G})$.*

For each $i \in [\tau]$ and $u \in V(\mathcal{G})$, denote by $C_i(u)$ the set of vertices in the same connected component of G_i as u, and by $R_i(u)$ the set of vertices in G_i reachable from u (i.e., $v \in R_i(u)$ if and only if there is a u, v-path in G_i). Observe that, if G is undirected, then $C_i(u) = R_i(u)$. For the sake of completeness, we now show that we can recursively define the set of vertices reachable from u by a temporal path finishing at time at most i. We apply the following lemma in the context of non-strict reachability, but also holds for strict.

Lemma 2. *Let $\mathcal{G}$ be a (directed) temporal graph, and let $\mathcal{R}_i(u)$ be recursively defined as:*

$$\mathcal{R}_i(u) = \begin{cases} R_1(u) & \text{, if } i = 1 \\ \bigcup_{v \in \mathcal{R}_{i-1}(u)} R_i(v) & \text{, otherwise} \end{cases}$$

Then $\mathcal{R}_i(u)$ is equal to the set of vertices reachable from u by a temporal path finishing at time at most i.

Proof. We want to prove that $v \in \mathcal{R}_i(u)$ if and only if there exists a temporal u, v-walk finishing in time at most i. First, let $v \in \mathcal{R}_i(u)$. If $i = 1$, then $v \in R_1(u)$ and u reaches v in G_1 by definition. So suppose $i > 1$. Again by definition, we have $v \in \bigcup_{v' \in \mathcal{R}_{i-1}(u)} R_i(v')$. Consider then $w \in \mathcal{R}_{i-1}(u)$ such that $v \in R_i(w)$. By induction hypothesis, there exists a temporal u, w-path P finishing in time at most $i - 1$. And because w reaches v in G_i, such path can be extended to a temporal u, v-walk finishing in time at most i.

Now, let v be such that there exists a temporal u, v-path P finishing in time at most i. If P finishes in time at most $i - 1$, we are done by induction hypothesis. Otherwise, let $w \in V(P)$ be closest to v in P such that the temporal edges incident to w in P occur in time $j < i$ and i. Observe that wPv is contained in G_i, and hence $v \in R_i(w)$. Additionally, uPw finishes in time at most $j \leq i - 1$, and by induction hypothesis $w \in \mathcal{R}_{i-1}(u)$. By definition we then get $v \in \mathcal{R}_i(u)$, as we wanted to show. □

The following easy proposition tells us that deciding the existence of large components (i.e. maximal connected sets) is equivalent to decide the existence of large connected sets.

Remark 1. Let $\mathcal{G}$ be a temporal (directed) graph. Then $\mathcal{G}$ has a (closed) temporal (unilateral) connected component of size at least k if and only if it has (closed) temporal (unilateral) connected set of size at least k.

3. Parameterized complexity results

This section is devoted to answer Question 1 and prove the results summarized in Table 1.

3.1. Parameterization by τ

We start by proving the result in the first column of Table 1, about para-NP-completeness when parameterized by the lifetime τ, which applies to all the definitions of temporal components. For (CLOSED) TCC we present a reduction from the NP-complete problem MAXIMUM EDGE BICLIQUE (MEBP for short) [47]. A *biclique* in a graph G is a complete bipartite subgraph of G. The MEBP problem consists of, given a bipartite graph G and an integer k, deciding whether G has a biclique with at least k edges. Using the same construction, we prove hardness of (CLOSED) TUCC reducing from the NP-complete problem $2K_2$-FREE EDGE SUBGRAPH [48]. In this problem we are given a bipartite graph G and an integer k, and are trying to decide whether G has a $2K_2$-free subgraph with at least k edges.

The main idea of the reductions is to generate a temporal graph $\mathcal{G}$ whose underlying graph is the line graph L of a bipartite graph H with parts X, Y. Recall that, for each $u \in X \cup Y$, there is a clique in L formed by all the edges incident to u; denote such clique by C_u. We make active in timestep 1 the edges within C_u for every $u \in X$, and in timestep 2 the edges within C_u for every $u \in Y$. Doing so, we ensure that any pair of vertices of $\mathcal{G}$ associated with a biclique in H reach one another in $\mathcal{G}$. We prove that there exists a biclique in H with at least k edges if and only if there exists a closed TCC in $\mathcal{G}$ of size at least k. The result extends to TCCs, as every TCC is also a closed TCC. For the unilateral case, we can relax the biclique to a $2K_2$-free graph since only one reachability relation is needed. As a result, we get the following.

Theorem 1. *For every fixed $\tau \geq 2$ and given a temporal graph $\mathcal{G} = (G, \lambda)$ of lifetime τ and an integer k, it is NP-complete to decide if $\mathcal{G}$ has a (closed) TCC or a (closed) TUCC of size at least k, even if G is the line graph of a bipartite graph.*

Proof. Consider an instance (H, k) of MEBP, consisting of a bipartite graph $H = (X \cup Y, E)$ and an integer k. We construct a temporal graph $\mathcal{G} = (G, \lambda)$ with lifetime 2 such that $V(G) = E(H)$ as follows. For each $x \in X$ and every pair of edges $e, e' \in \delta(x)$, add the edge ee' to $E(G)$, defining $\lambda(ee') = \{1\}$. Similarly, for each $y \in Y$ and every pair of edges $e, e' \in \delta(y)$, add the edge ee' to $E(G)$, defining $\lambda(ee') = \{2\}$. Clearly G is the line graph of H. We claim that there exists a biclique (A, B) in H with at least k edges if and only if there exists a closed TCC in $\mathcal{G}$ of size at least k. Then we prove that every TCC is also a closed TCC, finishing this part of the proof.

Suppose first that there exists a biclique (A, B) in H with at least k edges, and let $C = E(H[A, B])$. We want to show that C is a closed temporal connected set of $\mathcal{G}$. Let $e = xy$ and $e' = x'y'$ be two elements of C, with $\{x, x'\} \subseteq X$ and $\{y, y'\} \subseteq Y$. If $x = x'$ then $e, e' \in \delta(x)$ and hence are contained in the same component of G_1 (i.e., they reach each other by a direct edge); the analogous holds in case $y = y'$, so suppose $x \neq x'$ and $y \neq y'$. Since (A, B) is a biclique in H, we have that $\{xy, xy', x'y, x'y'\} \subseteq C$. Denote xy' by f and $x'y$ by f'. Now, in $\mathcal{G}$ we can reach f from e at timestep 1 and e' from f at timestep 2. Similarly, we can also reach e from e' in $\mathcal{G}$. Because f, f' are also in C, and since this holds for any two such edges, we get that C is a closed temporal connected set, and by Remark 1, we get that $\mathcal{G}$ has a closed TCC of size at least k.

For the converse, suppose that $\mathcal{G}$ has a TCC C with $|C| \geq k$. We want to show that C forms a biclique in H with at least k edges. Let $A \subseteq X$ contain all vertices incident to some $e \in C$, and define B similarly with relation to Y. First we show that (A, B) is a biclique in H. Let $x \in A$ and $y \in B$. We need to show that xy is an edge of H. Note that since $x \in A$, it must be an endpoint of some edge $e_x \in C$; analogously, since $y \in B$, it must be an endpoint of some edge $e_y \in C$. Since e_x and e_y are in C, there exists a temporal e_x, e_y-path in $\mathcal{G}$. Recall that the connected components of G_1 and of G_2 are cliques, and more specifically, that e_x is contained in the clique of G_1 formed by $\delta(x)$ and e_y is contained in the clique of G_2 formed by $\delta(y)$. Observe that this implies that if there exists a temporal e_x, e_y-path in $\mathcal{G}$, then there exists such a path of length at most 2. We analyse all cases. If $e_x e_y \in E(G_1)$, then $e_y \in \delta(x)$ and therefore $e_y = xy \in E(H)$ as we wanted to prove. A similar argument holds in case $e_x e_y \in E(G_2)$. Finally suppose that the temporal e_x, e_y-path in $\mathcal{G}$ is equal to $(e_x, 1, e, 2, e_y)$. Then $e \in \delta(x) \cap \delta(y)$, i.e., $e = xy \in E(H)$, as we wanted to prove. Finally just observe that $|E(H[A, B])| \geq |C| \geq k$ to see that we have the desired biclique in H.

Observe that we have proved that if H has a biclique with at least k edges, then $\mathcal{C}$ has a closed temporal connected set of size at least k, which means that $\mathcal{C}$ has a temporal connected set of size at least k. Then we proved that if $\mathcal{G}$ has a temporal connected set of size at least k (independently of it being closed or not), then H has a biclique with at least k edges. This finishes the proofs for (CLOSED) TCC.

For (CLOSED) TUCC, we make a reduction from $2K_2$-FREE EDGE SUBGRAPH on bipartite graphs. This was shown to be equivalent to the MINIMUM FILL-IN problem in co-bipartite graphs in [48], where the authors also showed that this problem is NP-complete. Given a bipartite graph H, the proof follows similarly to the first case, using exactly the same construction for the temporal graph $\mathcal{G}$. The proof is similar to the previous one, but we present it for completeness.

Suppose first that there exists a $2K_2$-free subgraph $H' = (A, B)$ in H with at least k edges, and let $C = E(H[A, B])$. We want to show that C is a closed temporal unilateral connected set of $\mathcal{G}$. Let $e = xy$ and $e' = x'y'$ be two elements of C, with $\{x, x'\} \subseteq X$ and $\{y, y'\} \subseteq Y$. If $x = x'$ then $e, e' \in \delta(x)$ and hence are contained in the same component of G_1 (i.e., they reach each other by a direct edge); the analogous holds in case $y = y'$, so suppose $x \neq x'$ and $y \neq y'$. Since H' is $2K_2$-free, we have that $\{xy', x'y\} \cap C \neq \emptyset$. If $f = xy' \in C$, then $(e, 1, f, 2, e')$ is a temporal e, e'-path in $\mathcal{G}$. And if $f' = x'y \in C$, then $(e', 1, f', 2, e)$ is a temporal e', e-path in $\mathcal{G}$. In either case, we have that e temporally reaches e' or the other way around. Since this holds for any two edges in C, we get that C is a closed temporal unilateral connected set, and by Remark 1, we get that $\mathcal{G}$ has a CLOSED TUCC of size at least k.

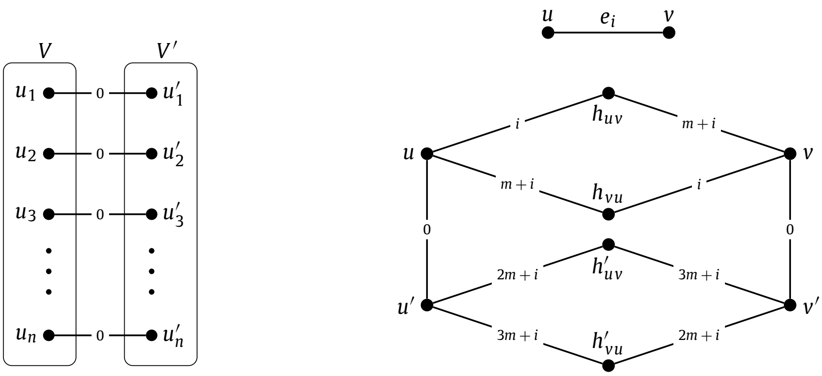

Fig. 3. Construction used in the proof of Theorem 2. On the left, the two copies of $V(G)$ and the edges between them, active in timestep 0. On the right, the edge $e_i \in E(G)$ and the associated gadget in $\mathcal{G}$.

For the converse, suppose that $\mathcal{G}$ has a temporal unilateral connected set C with $|C| \geq k$. We want to show that C forms a $2K_2$-free subgraph in H with at least k edges. Let $A \subseteq X$ contain all vertices incident to some $e \in C$, and define B similarly with relation to Y. Let H' be the subgraph of H induced by $A \cup B$. Since $|E(H')| \geq |C| \geq k$, it suffices to show that H' is $2K_2$-free. So let $e = xy, e' = x'y' \in E(H')$ be a pair of independent edges in H', i.e., $\{x, y\} \cap \{x', y'\} = \emptyset$. We need to prove that $\{xy', x'y\} \cap E(H') \neq \emptyset$. Because C is a temporal unilateral connected set, we can suppose, without loss of generality, that e temporally reaches e'. Since e, e' are not incident in the same vertex, we get that there must exist a temporal e, e'-path of length exactly two, say $(e, 1, f, 2, e')$. By construction, we get that $f \in \delta(x) \cap \delta(y')$; hence $f = xy'$. As this holds for every choice of such edges of H', it follows that H' is indeed $2K_2$-free.

Observe that we have proved that if H has a $2K_2$-free subgraph with at least k edges, then $\mathcal{C}$ has a closed temporal unilateral connected set of size at least k, which means that $\mathcal{C}$ has a temporal unilateral connected set of size at least k. Then we proved that if $\mathcal{G}$ has a temporal unilateral connected set of size at least k (independently of it being closed or not), then H has a $2K_2$-free subgraph with at least k edges. This finishes the proofs for (closed) tucc.

In order to extend these results for temporal graphs with higher lifetime, observe that we can artificially increase the lifetime to τ, for any fixed τ, simply by adding two new vertices u, v, together with edge uv, defining $\lambda(uv) = [\tau]$. □

3.2. W[1]-hardness by k

We now focus on proving the W [1]-hardness results in the second column of Table 1 concerning parameterization by k, which also imply some of the results of the third column. The following W [1]-hardness results (Theorem 2, 4, and 3) are parameterized reductions from k-Clique. The general objective is constructing a temporal graph $\mathcal{G}$ in a way that vertices in $\mathcal{G}$ are in the same component if and only if the corresponding nodes in the original graph are adjacent. Notice that we have to do this while: (i) ensuring that the size of the desired component is $f(k)$ for some computable function k (i.e., this is a parameterized reduction); and (ii) avoiding that the closed neighborhood of a vertex forms a component, so as to not a have a false "yes" answer to k-Clique. To address these tasks, we rely on different techniques. The first reduction concerns tcc in undirected temporal graphs and requires τ to be unbounded, as for τ bounded we show that the problem is FPT by $k + \tau$ (Theorem 5). The technique used is similar to the *semaphore* technique [12,17], except that it produces a parameterized reduction. While the original reduction gives labels in order to ensure that paths longer than one are broken, the following one allows the existence of paths longer than one. But if a temporal path from u to v exists for $uv \notin E(G)$, then the construction ensures the non-existence of temporal paths from v to u. Because of this property, the reduction does not extend to tucc, which we prove to be FPT when parameterized by k instead (Theorem 5).

Theorem 2. *Given a temporal graph $\mathcal{G}$ and an integer k, deciding if $\mathcal{G}$ has a TCC of size at least k is W[1]-hard with parameter k.*

Proof. We make a parameterized reduction from k-Clique. Let G be graph and $k \geq 3$ be an integer. We construct the temporal graph $\mathcal{G} = (G', \lambda)$ as follows. Fist, add to G' every vertex in $V(G)$ and make $V = V(G)$. Second, add to G' a copy u' of every vertex $u \in V$ and define $V' = \{u' \mid u \in V\}$. Third, for every pair u, u' with $u \in V$ and $u' \in V'$ add the edge uu' to G' and make all such edges active at timestep 0. Fourth, consider an arbitrary ordering $e_1, \ldots, e_m$ of the edges of G and for each edge $e_i = uv$ create four new vertices $\{h_{uv}, h_{vu}, h'_{uv}, h'_{vu} \mid uv \in E(G)\}$. We say that vertices $h_{uv}, h_{vu}, h'_{uv}, h'_{vu}$ are *related to edge* uv. Finally, add edges:

- uh_{uv} and vh_{vu}, active at time i;
- $u'h'_{uv}$ and $v'h'_{vu}$, active at time $2m + i$;
- $h_{vu}u$ and $h_{uv}v$, active at time $m + i$; and
- $h'_{vu}u'$ and $h'_{uv}v'$, active at time $3m + i$.

See Fig. 3 for an illustration of this construction. Denote the set $\{h_{uv}, h_{vu} \mid uv \in E(G)\}$ by H, and the set $\{h'_{uv}, h'_{vu} \mid uv \in E(G)\}$ by H'. We now prove that G has a clique of size at least k if and only if $\mathcal{G}$ has a temporal connected set of size at least $2k$. The theorem follows by Remark 1.

First, let $C \subseteq V$ be a clique of size at least k in G and $C' = \{u' \mid u \in C\}$. We show that $C \cup C'$ is a TCC of $\mathcal{G}$. For this, let $u, v \in C$. Since $\lambda(uh_{uv}) = i < m + i = \lambda(h_{uv}v)$, we get that u reaches v in $\mathcal{G}$ through h_{uv}. Because C is a clique in G we conclude that C is a temporal connected set of $\mathcal{G}$, and similarly the same holds for C'. Thus it remains to show that pairs of vertices of the form u, u' with $u \in C$ and $u' \in C'$ are also connected in $\mathcal{G}$. This is true due to the choice of timestep 0 for the edges forming the matching between V and V' of $\mathcal{G}$.

Now, let $S \subseteq V(G')$ be a TCC of $\mathcal{G}$ of size at least $2k$. We want to show that either $C = \{u \in V(G) \mid u \in S \cap V\}$ or $C' = \{u \in V(G) \mid u' \in S \cap V'\}$ is a clique of G of size at least k. For this, we first prove a series of useful facts.

Claim 1. *Let P be a temporal path in $G'[V \cup H]$. Then P has at most one internal vertex of H, and hence $|V(P)| \leq 5$. The analogous holds if P is contained in $G'[V' \cup H']$.*

Proof. Observe that every $e \in H$ is incident to exactly two edges of G', one active at time at most m and the other one active at time at least $m + 1$. This immediately gives that P contains at most one internal vertex of H. Additionally, since $G'[V \cup H]$ is a bipartite graph and P cannot contain more than one internal vertex of H, we get that P is the longest if it starts and finishes in H, having exactly one internal vertex of H. This implies that $|V(P)| \leq 5$ as desired. □

Claim 2. *C and C' are cliques in G.*

Proof. Let $u, v \in C$. Since S is a temporal connected set and $u, v \in C \subseteq S$, there is a temporal path from u to v. Such path must contain only edges of $G'[V \cup H]$ since the edges between V and V' are only active in timestep 0, and all other edges are active in a later time (i.e., there is no way to leave u to u' at time 0, then go back to v). By Claim 1 and the fact that $G'[V \cup H]$ is bipartite, it follows that u and v must be adjacent. The argument for $u, v \in C'$ is analogous by taking their copies, u', v' in S. □

Note that if $S \subseteq V \cup V'$, then Claim 2 and the fact that $|S| \geq 2k$ directly imply that either C or C' is a clique of size at least k in G. Assume now that $S \cap (H \cup H') \neq \emptyset$. In this case, it is not ensured that C or C' contains a clique of size at least k, but the following claims allow us to obtain another clique.

Claim 3. *For every $h \in H$ and every $x' \in V' \cup H'$, h does not reach x'. Similarly, for every $h' \in H'$ and every $x \in V \cup H$, h' does not reach x.*

Proof. The only edges between $V \cup H$ and $V' \cup H'$ are those incident to V and V' at timestep 0. Since every edge incident to $h \in H \cup H'$ is active only at a later timestep, the claim follows. □

Claim 4. *If $a, b \in S \cap H$, then a and b are related to the same edge, or to edges adjacent to each other. The same holds for $a, b \in S \cap H'$.*

Proof. Suppose, without loss of generality, that a reaches b. Suppose also by contradiction that a, b are related to distinct edges, say e_i and e_j, respectively. Write e_i as uv and e_j as xy and assume that $\{u, v\} \cap \{x, y\} = \emptyset$. Because every temporal path between a and b must alternate between V and H, as $G'[V \cup H]$ is bipartite, and since by Claim 1 every temporal path contains at most one internal vertex of H, we get that the temporal a, b-path must use vertices (a, v, h_{vx}, x, b). This gives us that av and vh_{vx} must be active in timestep at most m, while $h_{vx}x$ and xb must be active in timestep at least $m + 1$. Hence, by letting vx be equal to e_ℓ, we must have that $i < \ell < j$. We apply an analogous argument to a temporal b, a-path to obtain that j must be smaller than i, a contradiction. A similar argument can clearly be applied to $e, f \in S \cap H'$, and the claim follows. □

Now suppose that $S \cap H \neq \emptyset$. By Claim 3 we get that $S \subseteq V \cup H$. Since V and H are disjoint and $|S| \geq 2k$, we get that either $|S \cap V| \geq k$ or $|S \cap H| \geq k$. If the former occurs, then C contains a clique of size at least k by Claim 2. Otherwise, denote by E_S the set of edges of G related to vertices in $S \cap H$ (i.e. $E_S = \{uv \in E(G) \mid \{h_{uv}, h_{vu}\} \cap S \neq \emptyset\}$). The following is the last ingredient of the proof.

Claim 5. *Let $a, b \in S \cap H$ be related with distinct edges g, g' of G sharing an endpoint v. If u and w are the other endpoints of g and g', respectively, then u and w are also adjacent in G. Additionally, either $|S \cap \{h_{xy}, h_{yx}\}| \leq 1$ for every $xy \in E(G)$, or $|S \cap H| \leq 2$.*

Proof. By contradiction suppose that u and w are not adjacent in G. This gives us that every uw-path in G' contains two internal vertices of H, and therefore is not a temporal path by Claim 1. Because every subpath of a temporal path is also a temporal path, this means that there is no temporal a, b-path passing by u and w. By construction, and since G is a simple graph (i.e., there is only one edge with endpoints u and v, and only one with endpoints v and w), we get from Claim 1 that, if P is a temporal a,b-path not containing both u and w, then P is one of the following paths: $P_1 = (a, v, b)$; $P_2 = (a, u, a', v, b)$ where $\{a, a'\} = \{h_{uv}, h_{vu}\}$; or $P_3 = (a, v, b', w, b)$ where $\{b, b'\} = \{h_{vw}, h_{wv}\}$. Note that, since all paths are strictly increasing, we get that at least one between P_2 or P_3 (or their reverse) is a temporal path. But observe that neither the subpath (a, u, a', v) nor its reverse can ever be temporal paths by construction, which means that neither P_2 nor its reverse can be temporal paths. A similar argument can be applied to P_3, which means that the only possible temporal a,b-path is P_1. Observe that this in turn implies that edge av is active in a timestep strictly smaller than the timestep in which vb is active. Now, note that the same argument can be applied to conclude that the only possible temporal b,a-path is (b, v, a), a contradiction (recall that both paths must exist as $\{a, b\} \subseteq S$).

For the second part, suppose by contradiction that $\{h_{xy}, h_{yx}\} \subseteq S$ and $|S \cap H| > 2$. Let $a \in (S \cap H) \setminus \{h_{xy}, h_{yx}\}$. By Claim 4 we can suppose, without loss of generality, that $a \in \{h_{xw}, h_{wx}\}$ for some $w \neq y$. Observe also that the previous paragraph tells us that one of the temporal paths between $\{h_{xy}, h_{yx}\}$ and a must contain (y, f, w) or its reverse, where $f \in \{h_{yw}, h_{wy}\}$. Since such a path contains 4 edges, by letting xy be equal to e_i, yw be equal to e_j and wx be equal to e_ℓ, we get $i < j < \ell$. Thus in this case we have that wf is active in time at least $m + 1$, which in turn gives us that $a = h_{xw}$. We can now verify that a does not reach h_{xy}. Indeed, every a, h_{xy}-path starting with edge aw must contain some internal vertex h of H, in which case it cannot be a temporal path as it starts with an edge active at time at least $m + 1$ (namely aw) and contains an edge active in time at most m (namely one of the edges incident to h). A similar argument can be applied if the path starts with edge ax, since it must be distinct from (e, x, h_{xy}) (recall that $\lambda(ax) = \ell > i = \lambda(xh_{xy})$). □

Now, recall that we are in the case $|S \cap H| \geq k + 1$. By our assumption that $k \geq 3$, note that Claim 5 gives us that $|S \cap \{h_{xy}, h_{yx}\}| \leq 1$ for every $xy \in E(G)$, which in turn implies that $|E_S| = |S \cap H|$. Additionally, observe that, since $|S \cap H| \geq 4$, Claim 5 also gives us that there must exist $w \in V$ such that e is incident to w for every $e \in E_S$. Indeed, the only way that 3 distinct edges can be mutually adjacent without being all incident to a same vertex is if they form a triangle. Supposing that 3 edges in E_S form a triangle $T = (a, b, c)$, since $|E_S| \geq 4$, there exists an edge $e \in E_S \setminus E(T)$. But now, since G is a simple graph, e is incident to at most one between a, b and c, say a. We get a contradiction to Claim 5 as in this case e is not incident to edge $bc \in E_S$. Finally, by letting $C'' = \{v_1, \ldots, v_k\}$ be any choice of k distinct vertices such that $\{wv_1, \ldots, wv_k\} \subseteq E_S$, Claim 5 gives us that v_i and v_j are adjacent in G, for every $i, j \in [k]$; i.e., C'' is a clique of size at least k in G. This finishes the proof as the case $S \cap H' \neq \emptyset$ is clearly analogous. □

The following result concerns TCC and TUCC in directed temporal graphs. It is important to remark that for TCC and τ unbounded, we already know that the problem is W [1]-hard because of Theorem 2 which holds for undirected graphs and extends to directed ones. However, the following reduction applies specifically for directed ones already for $\tau = 2$. The technique used here is the previously mentioned semaphore technique, made parameterized by exploiting the direction of the edges. Namely, we reduce from k-Clique by replacing every edge uv of G by two vertices h_{uv} and h_{vu} and the directed temporal paths $(u, 1, h_{uv}, 2, v)$ and $(v, 1, h_{vu}, 2, u)$. Fig. 4(b) shows the temporal graph obtained by applying the reduction to the graph in Fig. 4(a). One can check that G has a clique of size at least k if and only if $\mathcal{G}$ has a TCC of size at least k. For TUCC, we only need to add one of h_{uv} or h_{vu}.

Theorem 3. *Given a directed temporal graph $\mathcal{G}$ and an integer k, deciding if $\mathcal{G}$ has a TCC of size at least k is W[1]-hard with parameter k, even if $\mathcal{G}$ has lifetime 2. The same holds for TUCCs.*

Proof. As previously said, we make a reduction from k-Clique. See Fig. 4(b), which is a temporal graph obtained from the graph in Fig. 4(a), to follow the construction. Let G be a graph and consider the directed graph D_G constructed as follows. First, add to D_G every vertex of G. Then, for each $uv \in E(G)$, add to D_G vertices h_{uv} and h_{vu}, directed edges uh_{uv} and vh_{vu}, and directed edges $h_{uv}v$ and $h_{vu}u$. Denote by H the set $\{h_{uv}, h_{vu} \mid uv \in E(G)\}$. To construct the directed temporal graph $\mathcal{G}$ we start from D_G and for every $uv \in E(G)$

- make edges uh_{uv} and vh_{vu} active in timestep 1; and
- make edges $h_{uv}v$ and $h_{vu}u$ active in timestep 2.

Assume $k \geq 3$. We now prove that G has a clique of size k if and only if $\mathcal{G}$ has a temporal connected set of size at least k. The theorem follows by Remark 1. Notice that every vertex of G is contained in $V(\mathcal{G})$, and that $\mathcal{G}$ has lifetime 2.

If C is a clique in G, then for every $u, v \in C$, we get that u reaches v and v reaches u in $\mathcal{G}$ because of the paths $(u, 1, h_{uv}, 2, v)$ and $(v, 1, h_{vu}, 2, u)$. It remains to show that if $\mathcal{G}$ has a temporal connected set of size at least k, then G has a clique of size at least k. Let C' be such a temporal connected set. We prove that $C' \subseteq V(G)$ and $uv \in E(G)$ for every $u, v \in C'$. First observe that G_1 has only edges from $V(G)$ to H, and G_2, from H to $V(G)$. This implies that a temporal path must be of length at most 2. Observe also that if $uv \notin E(G)$, then every u, v-path in D_G has length at least 4 and hence is not a temporal path in $\mathcal{G}$. As C' is temporal connected, we get that $C' \cap V(G)$ must be a clique. Now suppose that there exists $h_{uv} \in C' \cap H$. Observe that h_{uv} has exactly one incoming edge, active in timestep 1, and exactly one outgoing edge, active in timestep 2. Additionally, observe that every edge outgoing from v is active in timestep 1. This means that v is the

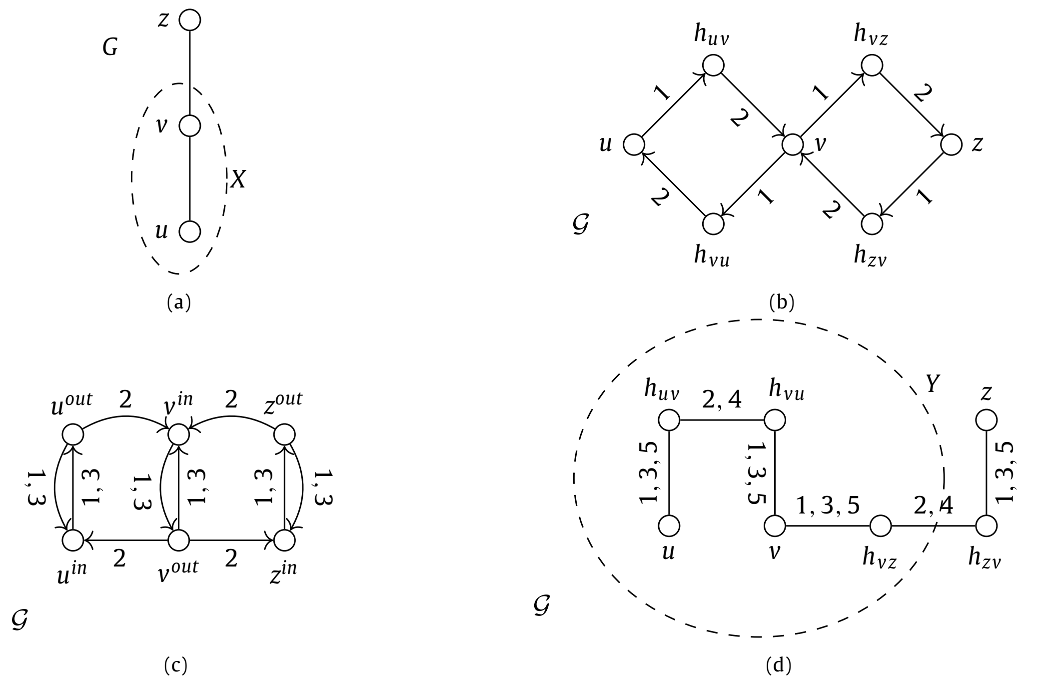

Fig. 4. Examples for some of our reductions. Given the graph in (a), Theorem 3 constructs the directed temporal graph in (b), Theorem 4 constructs the directed temporal graph in (c), and, given additionally set X in (a), Theorem 7 constructs the temporal graph $\mathcal{G}$ and set Y in (d). Sets X and Y are represented inside dashed circles.

only vertex of $V(\mathcal{G})$ reachable from h_{uv}, contradicting the fact that $k \geq 3$. Thus we conclude that $C' \subseteq V(G)$ and the result follows.

Now, for the unilateral case, observe that every TCC is also a TUCC, hence from the above paragraph we get that if G has a clique of size at least k, then $\mathcal{G}$ has a TUCC of size at least k. Now, if $\mathcal{G}$ has a TUCC of size at least k, then observe that the same arguments as before can be applied. Indeed, if $u, v \in C' \cap V(G)$, then it must be that either u reaches v or v reaches u, and in any case we have $uv \in E(G)$. Additionally, we know that C' cannot contain any vertex of H, as $k \geq 3$ and v is the only vertex reachable by h_{uv} for every $h_{uv} \in H$. □

The next result concerns closed TCCs and closed TUCCs. In this case, we also reduce from k-Clique, but we cannot apply the semaphore technique as before. Indeed, as we are dealing with closed components, nodes must be reachable using vertices inside the components, while the semaphore technique would make them reachable via additional nodes, which do not necessarily reach each other. For this reason, in the following we introduce a new technique subdividing nodes, instead of edges, in order to break paths of the original graph of length longer than one, being careful to allow that these additional nodes reach each other. The construction is shown in Fig. 4, which shows how to construct temporal graph $\mathcal{G}$ in Fig. 4(c), given graph G in Fig. 4(a) in a way that graph G has a clique of size k if and only if $\mathcal{G}$ has a closed TCC (TUCC) of size at least $2k$.

Theorem 4. *Given a directed temporal graph $\mathcal{G}$ and an integer k, deciding if $\mathcal{G}$ has a closed TCC of size at least k is W[1]-hard with parameter k, even if $\mathcal{G}$ has lifetime 3. The same holds for closed TUCCs.*

Proof. Again, we make a reduction from k-Clique. Observe Fig. 4(c), which is the temporal graph obtained from the graph G in Fig. 4(a), to follow the construction. Let G be a graph and consider the directed graph D_G constructed as follows. For every $u \in V(G)$, add to D_G vertices u^{in} and u^{out}, an edge from u^{in} to u^{out}, and an edge from u^{out} to u^{in} (notice that each pair $u^{\text{in}}, u^{\text{out}}$ induce a cycle in $\mathcal{G}$). Then, for each edge $uv \in E(G)$, add to D_G an edge from u^{out} to v^{in} and an edge from v^{out} to u^{in}. The directed temporal graph $\mathcal{G} = (D_G, \lambda)$ is such that λ is defined as follows.

- For every $u \in V(G)$, make edges between u^{in} and u^{out} active in timesteps 1 and 3 in both directions; and
- For every $uv \in E(G)$, make the edges from u^{out} to v^{in} and from v^{out} to u^{in} active in timestep 2.

We now prove that G has a clique of size at least k if and only if $\mathcal{G}$ has a closed temporal connected set of size at least $2k$. The theorem follows by Remark 1. Notice that $\mathcal{G}$ has lifetime 3. Let C be a clique of size at least k in G, and let $C' = \{u^{\text{in}}, u^{\text{out}} \in V(\mathcal{G}) \mid u \in C\}$. We prove that, for every $u, v \in C$ with $u \neq v$, the set $\{u^{\text{in}}, u^{\text{out}}, v^{\text{in}}, v^{\text{out}}\}$ is a closed temporal connected set; note that this implies that C' itself is a closed temporal connected set, as desired. By construction, for every $w \in V(G)$ there are temporal paths from w^{in} to w^{out} and the other way around, in other words u^{in} reaches u^{out}, and vice-versa, and v^{in} reaches v^{out} and vice-versa. Moreover, u^{in} reaches v^{in} in $\mathcal{G}$ through the path $(u^{\text{in}}, 1, u^{\text{out}}, 2, v^{\text{in}})$. Observe that this also implies that u^{out} reaches v^{in}, and by symmetry, that both v^{in} and v^{out} reach u^{in}. Finally, note that the path $(u^{\text{in}}, 1, u^{\text{out}}, 2, v^{\text{in}}, 3, v^{\text{out}})$ implies that both u^{in} and u^{out} reach v^{out}, and by symmetry we also get that v^{in} and v^{out} reach u^{out}. This finishes this part of the proof.

Assume now that C' is a closed temporal connected set of $\mathcal{G}'$ of size at least $2k$. Let $C = \{u \in V(G) \mid \{u^{\text{in}}, u^{\text{out}}\} \cap C' \neq \emptyset\}$. Clearly $|C| \geq k$ since $|C'| \geq 2k$. To show that C is a clique in G, observe that $\mathcal{G}$ consists of a matching at timesteps 1 and 3, containing only edges of the form $u^{\text{in}}u^{\text{out}}$ and of the form $u^{\text{out}}u^{\text{in}}$, together with edges in timestep 2 that go only from $O = \{u^{\text{out}} \mid u \in V(G)\}$ to $I = \{u^{\text{in}} \mid u \in V(G)\}$. This implies that any temporal path in $\mathcal{G}$ contains at most one edge from O to I, which are only defined if the corresponding vertices are adjacent in G. We then get that, if $u, v \in C$ with $u \neq v$, then it must be the case that $uv \in E(G)$.

The proof for CLOSED TUCC is similar, except that, for every $uv \in E(G)$, we only need to add either $u^{\text{out}}v^{\text{in}}$ or $v^{\text{out}}u^{\text{in}}$ to $\mathcal{G}$. Formally, let $\mathcal{G} = (D_G, \lambda)$ be a directed temporal graph obtained as follows. For every $u \in V(G)$, add to D_G vertices u^{in} and u^{out}, an edge from u^{in} to u^{out}, and an edge from u^{out} to u^{in}; make all such edges active in timesteps 1 and 3. Then, choose any ordering $v_1, \ldots, v_n$ of $V(G)$ and, for each edge $v_iv_j \in E(G)$, $i < j$, add to D_G an edge from v_i^{out} to v_j^{in}, active in timestep 2. Even though the proof is quite similar to the previous one, we reproduce it here for completeness.

First, let C be a clique of size at least k in G, and let $C' = \{u^{\text{in}}, u^{\text{out}} \in V(\mathcal{G}) \mid u \in C\}$. Consider $v_i, v_j \in C$ with $i < j$. We have that $P_{i,j} = (v_i^{\text{in}}, 1, v_i^{\text{out}}, 2, v_j^{\text{in}}, 3, v_j^{\text{out}})$ is a temporal path. Note that this implies that $\{v_i^{\text{in}}, v_i^{\text{out}}, v_j^{\text{in}}, v_j^{\text{out}}\}$ is a closed temporal unilateral connected set, which in turn implies that C' is a closed temporal unilateral connected set, as desired. For the second part of the proof, the reader should observe that the exact same argument used for temporal connected sets holds also here. □

3.3. FPT algorithms

We now show our FPT algorithms to find (closed) TCCs and (closed) TUCCs in undirected temporal graphs, as for directed temporal graphs we have proved W [1]-hardness. In particular, we prove the result in Theorem 5. It is important to observe that for unilateral components, the bounds in Theorem 5 depend only on k, while for TCCs and closed TCCs they depend on both k and τ. This is consistent with the fact that we have proved that for TCCs the problem is W[1]-hard when parameterized just by k (Theorem 2).

The idea of the proof of Theorem 5 exploits the fact that finding a TCC (resp. TUCC) in $\mathcal{G}$ of size at least k is equivalent to finding a set $S \subseteq V(\mathcal{G})$ in the reachability graph $R = R(\mathcal{G})$ (see Section 2) of size *exactly* k such that $uv \in E(R)$ and (resp. or) $vu \in E(R)$ for every pair $u, v \in V(R)$. As for finding a closed TCC (resp. closed TUCC), we need to have the same property, except that all subsets of size *at least* k must be tested (recall that being a closed connected (unilateral) set is not hereditary). Therefore, if Δ is the maximum degree of R, then testing connectivity takes time $O(\Delta^k \cdot n)$ (it suffices to test all subsets of size $k-1$ in $N(u)$, for all $u \in V(R)$), while testing closed connectivity takes time $O(2^\Delta \cdot n)$ (it suffices to test all subsets of size *at least* $k-1$ in $N(u)$, for all $u \in V(R)$). The proofs then consist in bounding the value Δ in each case.

Theorem 5. *Given a temporal graph $\mathcal{G} = (G, \lambda)$ on n vertices and with lifetime τ, and a positive integer k, there are algorithms running in time*

1. $O(k^{k\cdot\tau} \cdot n)$ *that decides whether there is a TCC of size at least k;*
2. $O(2^{k^\tau} \cdot n)$ *that decides whether there is a closed TCC of size at least k;*
3. $O(k^{k^2} \cdot n)$ *that decides whether there is a TUCC of size at least k; and*
4. $O(2^{k^k} \cdot n)$ *that decides whether there is a closed TUCC of size at least k.*

Proof. Let $\mathcal{G}$ be a temporal graph and k be a positive integer. We first prove points 1 and 2. Denote by F the graph obtained from the reachability digraph $\mathrm{R}(\mathcal{G})$ by removing all edges that are not symmetric and taking the underlying graph. Formally, F is an undirected graph with vertex set $V(G)$ and edge set $\{uv \mid \{uv, vu\} \subseteq E(\mathrm{R}(\mathcal{G}))\}$. Lemma 1 and Remark 1 tell us that TCC is equivalent to finding a clique of size k in F. As for CLOSED TCC, observe that Lemma 1 tells us that if $\mathcal{G}$ has a closed TCC, then it must form a clique in F. Therefore, if all cliques of size at least k in F are not closed connected sets, by Remark 1 we can conclude that $\mathcal{G}$ does not have a closed TCC of size at least k. In other words, solving CLOSED TCC is equivalent to finding a clique S of size at least k in F such that $\mathcal{G}[S]$ is connected. Observe that if $\Delta(F) \leq \Delta$ for some value Δ, then the former can be solved by testing, for every $u \in V(F)$ and every $S \subseteq N_F(u)$ with $|S| = k-1$, whether $S \cup \{u\}$ is a clique in F; this takes time $O(\Delta^k \cdot k^2 \cdot n)$. Now for the latter, we need to test for the existence of such sets of bigger sizes. This is because closed TCCs are not closed under inclusion. Nevertheless, since $\Delta(G) \leq \Delta$ and testing whether $\mathcal{G}[S]$ is connected can be done in time $O(|S| \cdot |E(G[S])|)$, we can test for the existence of a closed TCC in time $O(2^\Delta \cdot \Delta^3 \cdot n)$ by searching all cliques of size at least k in $N[u]$, for every $u \in V(G)$. We finish the proof by bounding the value of Δ.

Now, we show that $\Delta \leq (k-1)^\tau$, which combined with the previous paragraph gives us the stated running time. For this, first notice that, for every $i \in [\tau]$, the vertex set of any connected component C of G_i is a clique in F and a closed temporal connected set of $\mathcal{G}$. This means that we can suppose that the size of any connected component of G_i is at most $k-1$, for every $i \in [\tau]$, as otherwise we have a trivial yes instance for both problems. Now, since $C_i(u) = R_i(u)$ when G is undirected, apply Lemma 2 to see that $|\mathcal{R}_\tau(u)| \le (k-1)^\tau$. Additionally, by definition we know that $\mathcal{R}_\tau(u)$ contains exactly the set of vertices reachable by u in $\mathcal{G}$. Since $v \in N_F(u)$ if and only if u reaches v *and* v reaches u, it follows that $N_F(u) \subseteq \mathcal{R}_\tau(u)$. This finishes the proof of items 1 and 2.

Now we turn our attention to items 3 and 4, namely, algorithms running in time:

- $O(k^{k^2} \cdot n)$ that decides whether there is a TUCC of size at least k; and
- $O(2^{k^k} \cdot n)$ that decides whether there is a closed TUCC of size at least k.

Again by applying Lemma 1, a similar argument as the one used for items 1 and 2 can be applied directly to the reachability graph $F = \mathrm{R}(\mathcal{G})$ to say that, if $\Delta(F) \le \Delta$, then TUCC can be solved in time $O(\Delta^k \cdot k^2 \cdot n)$, while CLOSED TUCC can be solved in time $O(2^\Delta \cdot \Delta^3 \cdot n)$. hence it remains to bound Δ.

We first bound the degree in G, namely we prove first that $d_G(u) \le k-2$ for every $u \in V(G)$. This holds because, given any pair $v, w \in N(u)$, and any choice of values $i \in \lambda(uv)$ and $j \in \lambda(uw)$, either we have $i \le j$, in which case (v, i, u, j, w) is a temporal path, or $i > j$, in which case (w, j, u, i, v) is a temporal path. In other words, for every $v, w \in N(u)$, either v reaches w or w reaches v, which implies that $N_G[u]$ is a clique in F, for every $u \in V(G)$. Hence if $d_G(u) \ge k-1$, we are in a trivial yes instance. To finish, just observe that any temporal path forms a closed TUCC, which is always contained in a TUCC. Therefore, we can suppose that any vertex reachable from u is reached by a temporal path containing at most $k-1$ edges. Because $d_G(v) \le k-2$ for every $v \in V(G)$, we get that $d_F(u) = |\mathcal{R}_\tau(u)| \le (k-2)^{k-1}$ and the result follows. □

4. Checking connectivity

This section is focused on Question 2, which is open for all definitions of components for both the strict and the non-strict models. We answer to the question providing the following conditional lower bound, which holds for both models, where the notation $\tilde{O}(\cdot)$ ignores polylog factors.

Theorem 6. *Consider a (directed) temporal graph $\mathcal{G}$ on M temporal edges. There is no algorithm running in time $\tilde{O}(M^{2-\epsilon})$, for some ϵ, that decides whether $\mathcal{G}$ is temporal (unilateral) connected, unless* SETH *fails. This holds for the strict and non-strict models.*

We apply the technique used for instance in [49–51] to prove lower bounds for polynomial problems, falling within the fine-grained complexity framework. We use *quasilinear Karp reductions*, i.e. Karp reductions running in quasilinear time, whose formal definition is given in [49]. Intuitively, it consists of a Karp reduction where the new instance I' obtained from I has size at most $\tilde{O}(|I|)$ and can be obtained also in such time. We recall the reader that $\tilde{O}(\cdot)$ neglects poly-logarithmic factors, i.e., $\tilde{O}(f(n))$ equals $O(f(n)\log^k n)$ for some fixed k.

The key idea is to reduce a starting problem that is known not to be solvable in subquadratic time to our problem using such kind of reduction. This seed problem is the following formulation of the k-SAT* problem. Let ϕ be a CNF formula on variables $\mathcal{X} = \{x_1, \dots, x_n\}$ and $\mathcal{Y} = \{y_1, \dots, y_n\}$, with m clauses of size at most k. Let X denote the set of all 2^n possible truth assignments for $\mathcal{X}$, and similarly let Y denote the set of all 2^n possible truth assignments for $\mathcal{Y}$. In the k-SAT* problem, given $I = (\phi, X, Y)$, the goal is to decide if ϕ is satisfiable. The main difference with relation to the classical k-SAT problem is the size of the input, which is $|I| = \Theta(2^n)$.

Remark 2 *([49]).* k-SAT* with input $I = (\phi, X, Y)$ as before cannot be solved in time $O(|I|^{2-\epsilon})$ for some ϵ, even if ϕ has $\log^k n$ clauses, unless SETH fails.

By presenting a quasilinear Karp reduction from k-SAT*, and applying Remark 2, we obtain that, unless SETH fails, there is no subquadratic-time algorithm that decides if a given temporal graph is temporal (unilateral) connected.

In this section, it is helpful to formally define the following two problems.

Problem TEMPORAL CONNECTED.
Input: A (directed) temporal graph $\mathcal{G}$.
Question: Is $\mathcal{G}$ temporal connected?

Problem TEMPORAL UNILATERAL CONNECTED.
Input: A (directed) temporal graph $\mathcal{G}$.
Question: Is $\mathcal{G}$ temporal unilateral connected?

For both problems, given an instance $I = (\phi, X, Y)$ of k-SAT*, we construct a temporal graph $\mathcal{G} = (G, \lambda)$ such that $\mathcal{G}$ is not temporal (unilateral) connected if and only if ϕ has a satisfying assignment. As in the obtained temporal graph all temporal paths are strict temporal paths, the result holds on both models, strict and non-strict. Additionally, we construct an undirected temporal graph and get the result for directed thanks to the following straightforward proposition.

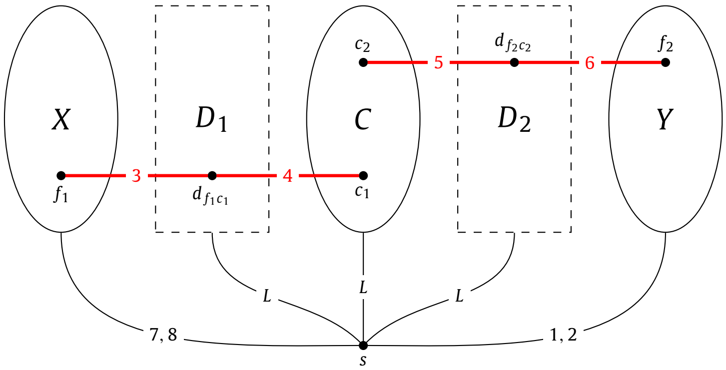

Fig. 5. General structure of the constructed graph in the reduction for the temporal connectivity testing problem. A black edge denotes the existence of all possible edges. A thick red edge denotes the temporal edges whose existence is conditioned to the assignment not satisfying the clause. L denotes the set of labels $\{1, 2, 7, 8\}$.

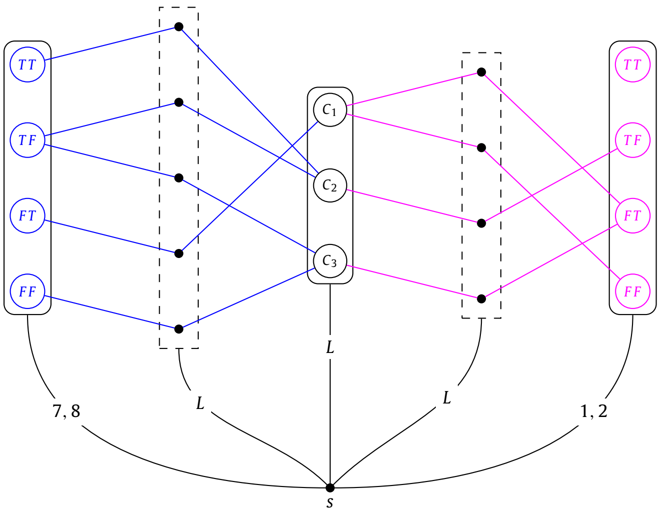

Fig. 6. Graph in the reduction for the temporal connectivity testing problem, related to the formula $\phi = (x_1 \vee \neg x_2 \vee y_1) \wedge (\neg x_1 \vee \neg y_1 \vee y_2) \wedge (x_2 \vee y_1 \vee \neg y_2)$. Blue nodes denote assignments of x_1 and x_2 (e.g., node TT blue denotes the assignment $x_1 = True$ and $x_2 = True$), while magenta nodes denote assignments of y_1 and y_2. Again, black edges denote existence of all possible edges. We put them outside the vertices in order to make the figure clean. Blue edges are active in times 3 and 4 (from left to right) and magenta edges, in times 5 and 6 (again from left to right). Also, L denotes the set of labels $\{1, 2, 7, 8\}$.

Proposition 1. *Let $\mathcal{G} = (G, \lambda)$ be a temporal graph and $\mathcal{G}' = (G', \lambda')$ be obtained from $\mathcal{G}$ by replacing each edge uv by two directed edges uv and vu with $\lambda'(uv) = \lambda'(vu) = \lambda(uv)$. Then $S \subseteq V(G)$ is temporal (unilateral) connected in $\mathcal{G}$ if and only if S is temporal (unilateral) connected in $\mathcal{G}'$.*

We first present a reduction from k-SAT* to the complement of Temporal Connected. See Fig. 5 to follow the construction. Let C be the set of clauses in ϕ. Let $V(G) = X \cup C \cup Y \cup D_1 \cup D_2 \cup \{s\}$, where D_1, D_2 are constructed as follows. For each $f \in X$ and $c \in C$ such that f does *not* satisfy c, add d_{fc} to D_1. Also add edges fd_{fc}, active at time 3, and $d_{fc}c$, active at time 4. Similarly, for each $f \in Y$ and $c \in C$ such that f does *not* satisfy c, add d_{fc} to D_2 and add edges cd_{fc}, active at time 5, and $d_{fc}f$, active at time 6. Denote by D the set $D_1 \cup D_2$. Finally, add all edges between s and $C \cup D$, active at times 1, 2, 7, 8, all edges between s and X, active at times 7, 8, and all edges between s and Y, active at times 1, 2. See Fig. 6 for an example.

We now argue that this is a quasilinear Karp reduction. Observe first that $|X| = |Y| = 2^n+$, $|C| = m$, and $|D| \leq m \cdot 2^{n+1}$; hence, $|V(G)| = O(m \cdot 2^n)$. Additionally, there are at most $m \cdot 2^{n+2}$ temporal edges non incident to s and at $4 \cdot |V(G) \setminus \{s\}|$ edges incident to s, totalling also $O(m \cdot 2^n)$ edges. Since Remark 2 tells us that m can be considered to be at most $\log^k n$, we get that the new instance has size $\tilde{O}(2^n) = \tilde{O}(|I|)$. It remains to prove correctness. Before we do that, we first argue that the reachability graph of $\mathcal{G}$ always contains uv for every u and v such that either $u \notin X$ or $v \notin Y$. For this, we analyse all cases below:

- s reaches v and is reachable from v, for every $v \in V(G) \setminus \{s\}$ by a direct edge;

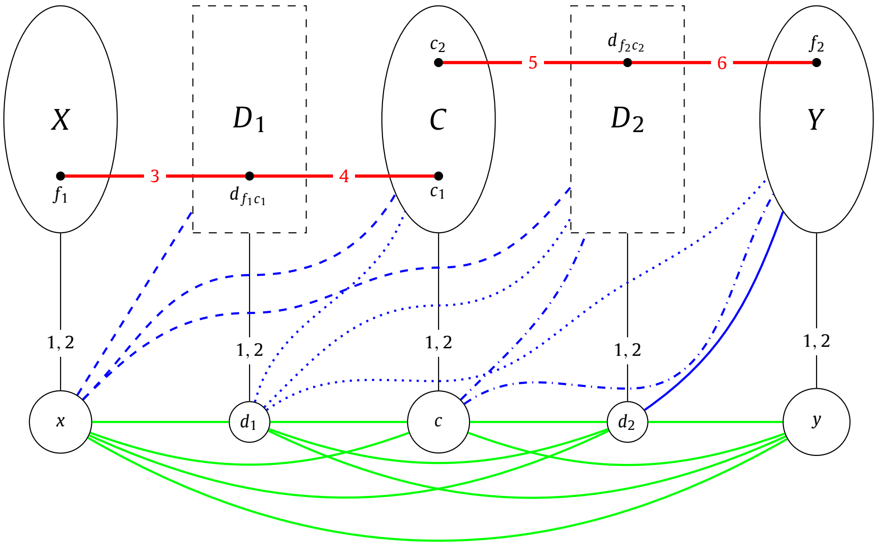

Fig. 7. General structure of the constructed graph in the reduction for the unilateral temporal connectivity testing problem. A black or blue edge denotes the existence of all possible edges. A red edge denotes the temporal edges whose existence is conditioned to the assignment not satisfying the clause. Blue edges are represented in different styles so it is possible to better recognize its endpoints.

- u reaches v for every $u, v \in C \cup D \cup X$: this holds for both strict and non-strict model using the temporal path $(u, 1, s, 2, v)$;
- u reaches v for every $u, v \in C \cup D \cup Y$: this holds for both strict and non-strict model using the temporal path $(u, 7, s, 8, v)$;
- y reaches x for every $y \in Y$ and $x \in X$: this holds for both strict and non-strict model using the temporal path $(y, 1, s, 7, x)$.

Observe then the only possible missing reachable pairs are of the form (x, y) where $x \in X$ and $y \in Y$, i.e., $\mathcal{G}$ is *not* temporal connected if and only if there exists $x \in X$ and $y \in Y$ such that x does *not* reach y. We now prove that ϕ is satisfiable if and only if $\mathcal{G}$ is *not* temporal connected. First, suppose that ϕ is satisfiable and consider a satisfying assignment f of ϕ. Then let $x \in X$ be equal to f restricted to $\{x_1, \dots, x_n\}$ and $y \in Y$ be equal to f restricted to $\{y_1, \dots, y_n\}$. Observe that for every $c \in C$, either x satisfies c, and hence $(xd_{xc}, 3) \notin E^T(\mathcal{G})$, or y satisfies c, and hence $(d_{yc}y, 6) \notin E^T(\mathcal{G})$. Therefore, there are no temporal paths from x to y of the form $(x, 3, d_{xc}, 4, c, 5, d_{yc}, 6, y)$. This means that any path starting with temporal edge $(xd_{xc}, 3)$ for some c must eventually "go down" to s using a temporal edge active at time 7 or 8. However this cannot be a temporal path because all temporal edges incident to Y are active at time at most 6. A similar argument can be applied to argue that no path starting with edge xs can be a temporal x, y-path. On the other hand, suppose that x does not reach y for some pair $x \in X$ and $y \in Y$. This means that there are no temporal x, y-paths of the form $(x, 3, d_{xc}, 4, c, 5, d_{yc}, 6, y)$, which in turn implies that, for every $c \in C$, either x satisfies c or y satisfies c. It immediately follows that $x \cup y$ is a satisfying assignment for ϕ.

Consider now the complement of TEMPORAL UNILATERAL CONNECTED. We make a similar reduction. The idea behind the modification is that we split s into 5 vertices and allow the vertices to reach every other vertex "to their right", except for the pairs (x, y) with $x \in X$ and $y \in Y$. Observe Fig. 7 to follow the construction. Let $V(G) = X \cup C \cup Y \cup D_1 \cup D_2 \cup \{x, y, c, d_1, d_2\}$, where D_1 and D_2, as well as the temporal edges incident to $D_1 \cup D_2$ and $X \cup C \cup Y$, are constructed exactly as before. Then, for each pair (a, A) where $a \in \{x, y, c, d_1, d_2\}$ and $A \in \{X, Y, C, D_1, D_2\}$, add all edges between a and A and make them active in timesteps 1 and 2. Then, add the following edges, all of them active at timestep 7: all edges between x and $D_1 \cup D_2 \cup C$; all edges between d_1 and $C \cup D_2 \cup Y$; all edges between c and $D_2 \cup Y$; and all edges between d_2 and Y. Finally, let $\{x, y, c, d_1, d_2\}$ be a clique active at timestep 8.

Let $\mathcal{G} = (G, \lambda)$ be the constructed temporal graph. One can see that the number of vertices and edges increased by a constant factor when compared to the previous reduction, so again we have that $\mathcal{G}$ has size $\tilde{O}(|I|)$. It remains to prove that ϕ is satisfiable if and only if $\mathcal{G}$ is *not* unilateral temporal connected. As before, we first prove that the only missing pairs are of the type (x, y) with $x \in X$ and $y \in Y$. We analyse all the other cases first, as we did for temporal connected. But first observe that, for each $a \in \{x, y, c, d_1, d_2\}$ and each $A \in \{X, Y, C, D_1, D_2\}$, it holds that: (*) if a reaches some u through a path starting at time at least 7, then a' reaches u for every $a' \in A$. In this case, for simplicity, we write A reaches u. The arguments below are repetitive, but we add all of it for completeness. The idea is that the reachability graph will contain all arcs from a to b, if a and b belong to the same set, and all arcs from A to B whenever B appears to the right of A in Fig. 7, except possibly arcs from X to Y.

- x' reaches u for every $x' \in X \cup \{x\}$ and every $u \in V(G) \setminus Y$: first observe that $(x', 1, x, 2, u)$ is a temporal path in case $u \in X$. For all other cases, just recall observation (*) and observe that either $(xu, 7)$ or $(xu, 8)$ is a temporal edge in $\mathcal{G}$;
- d reaches u for every $d \in D_1 \cup \{d_1\}$ and every $u \in V(G) \setminus X$: first observe that $(d, 1, d_1, 2, u)$ is a temporal path in case $u \in D_1$. For all other cases, just recall observation (*) and observe that either $(du, 7)$ or $(du, 8)$ is a temporal edge in $\mathcal{G}$;
- c' reaches u for every $c' \in C \cup \{c\}$ and every $u \in V(G) \setminus (X \cup D_1)$: first observe that $(c', 1, c, 2, u)$ is a temporal path in case $u \in C$. For all other cases, just recall observation (*) and observe that either $(cu, 7)$ or $(cu, 8)$ is a temporal edge in $\mathcal{G}$;
- d reaches u for every $d \in D_2 \cup \{d_2\}$ and every $u \in V(G) \setminus (X \cup D_1 \cup C)$: first observe that $(d, 1, d_2, 2, u)$ is a temporal path in case $u \in D_2$. For all other cases, just recall observation (*) and observe that either $(du, 7)$ or $(du, 8)$ is a temporal edge in $\mathcal{G}$;
- y' reaches y'' for every $y', y'' \in Y \cup \{y\}$: as y is adjacent to every vertex in Y, we can suppose $y \notin \{y', y''\}$. Hence, $(y', 1, y, 2, y'')$ is a temporal y', y''-path, as desired.

The proof of correctness is analogous to the previous one. Denote by A the set $\{x, y, c, d_1, d_2\}$. First observe that any path with more than one edge using a temporal edge active at time 7 or 8 must finish with such edge. One can see that this implies that the only possible temporal paths between the sets X and Y must contain only vertices of $V(G) \setminus A$. Observe that the temporal edges incident in Y and not in A are active in timestep 6, while the edges incident in X and not in A are active in timestep 3. This implies that no temporal path from Y to X is possible. Now, we prove that $\mathcal{G}$ is *not* temporal connected if and only if ϕ is satisfiable, which, by previous argument, occurs if and only if there exist $x \in X$ and $y \in Y$ such that x does not reach y. The proof can be done as before, with the additional knowledge that any such path must go through D_1, C, D_2, in this order.

Now observe that we have made reductions from k-SAT* to the *complements* of our problems. However, since a subquadratic-time algorithm that solves the complement $\overline{\Pi}$ of a problem Π, also solves Π (indeed I is a positive instance of Π if and only if I is a negative instance of $\overline{\Pi}$), we get that Theorem 6 follows.

5. Checking maximality

We now focus on Question 3. We prove the results in the second column of Table 2, about the problem of deciding whether a subset of vertices Y of a temporal graph is a component, i.e. a maximal connected set. The question is open both for the strict and the non-strict model. We argued already in the introduction that this is polynomial for TCCs and TUCCs for both models. In the following we prove NP-completeness for closed TCCs and closed TUCCs on undirected graphs. The results extend to directed graphs as well thanks to Proposition 1.

Theorem 7. *Let $\mathcal{G}$ be a (directed) temporal graph, and $Y \subseteq V(\mathcal{G})$. Deciding whether Y is a closed TCC is NP-complete. The same holds for closed TUCCs.*

Proof. We reduce from the problem of deciding whether a subset of vertices X of a given a graph G is a maximal 2-club, where a 2-club is a set of vertices C such that $G[C]$ has diameter at most 2. This problem has been shown to be NP-complete in [52]. Let us first focus on the strict model. In this case, given G we can build a temporal graph $\mathcal{G}$ with only two snapshots, both equal to G. Observe that X is a 2-club in G if and only if X is a closed TCC in $\mathcal{G}$. Indeed, because we can take only one edge in each snapshot and $\tau = 2$, we get that temporal paths will always have length at most 2. This also extends to closed TUCCs by noting that all paths in $\mathcal{G}$ can be temporally traversed in both directions.

In the case of the non-strict model, the situation is more complicated as in each snapshot we can take an arbitrary number of edges resulting in paths arbitrarily long. We show the construction for closed TCCs in what follows.

As before, we make a reduction from the problem of, given a graph G and $X \subseteq V(G)$, deciding whether X is a maximal 2-club. Observe Fig. 4(d), which is obtained by applying the reduction to the graph in Fig. 4(a). We obtain $\mathcal{G}$ from G by subdividing each edge $uv \in E(G)$ twice, creating vertices h_{uv} and h_{vu}, with $\lambda(uh_{uv}) = \lambda(vh_{vu}) = \{1, 3, 5\}$, and $\lambda(h_{uv}h_{vu}) = \{2, 4\}$. Observe that the first vertex in the subscript of h_{xy} tells us which between x and y is adjacent to h_{xy}. Denote by H the set $\{h_{uv}, h_{vu} \mid uv \in E(G)\}$. We now prove that X is a maximal 2-club in G if and only if $Y = X \cup N_H(X)$ is a closed TCC in $\mathcal{G}$. In fact, we prove that:

1. If $X \subseteq V(G)$ is such that $G[X]$ has diameter at most 2, then $Y = X \cup N_H(X)$ is a closed temporal connected set in $\mathcal{G}$; and
2. If $Y \subseteq V(\mathcal{G})$ is a closed temporal connected set, then $X = Y \cap V(G)$ is such that $G[X]$ has diameter at most 2.

We argue that indeed 1 and 2 above imply what we want, i.e., that X is a maximal 2-club in G if and only if $Y = X \cup N_H(X)$ is a closed TCC in $\mathcal{G}$. Observe that, supposing that 1 and 2 hold, if X is a maximal 2-club, then $Y = X \cup N_H(X)$ must be a closed TCC. Indeed, if $Y \subset Y'$ and Y' is a closed connected set (i.e., Y is not maximal), then by 2 we get that $X' = Y' \cap V(G)$ has diameter 2. Since X' contains X, this contradicts the choice of X. Conversely, if Y is a closed TCC, then X must be a maximal 2-club, as otherwise we could apply 1 to get a closed connected set strictly containing Y.

We first prove 1. So, consider $X \subseteq V(G)$ such that $G[X]$ has diameter at most 2, and define Y as above. Let $u, v \in Y \cap V(G)$. If $uv \in E(G)$, then $(u, 1, h_{uv}, 2, h_{vu}, 3, v)$ and $(v, 1, h_{vu}, 2, h_{uv}, 3, u)$ witness that u reaches v and v

reaches u in $\mathcal{G}[Y]$. And if $uv \notin E(G)$, then, since $G[X]$ has diameter 2, let $w \in N(u) \cap N(v)$ in G. We get the paths: $(u, 1, h_{uw}, 2, h_{wu}, 3, w, 3, h_{wv}, 4, h_{vw}, 5, v)$ and $(v, 1, h_{vw}, 2, h_{wv}, 3, w, 3, h_{wu}, 4, h_{uw}, 5, u)$. Therefore, u reaches v and v reaches u in $\mathcal{G}[Y]$. Now, consider $u \in X \cap Y$ and $h \in H \cap Y$. Let $v \in X$ be such that $h \in N(v)$ (observe that v is uniquely defined). If $v = u$, then u and v are clearly connected, so suppose otherwise. Because X has diameter at most 2, there exists a u, v-path P in $G[X]$ of length at most 2, say (u, w, v), with possibly $w = v$. Then either $(u, 1, h_{uw}, 2, h_{wu}, 3, w, 3, h)$ is a temporal u, h-path, in case $w = v$, or $(u, 1, h_{uw}, 2, h_{wu}, 3, w, 3, h_{wv}, 4, h_{vw}, 5, v, 5, h)$ is a temporal u, h-path, in case $w \neq v$. One can check that the symmetric path between h and u ensures that also h reaches u in $\mathcal{G}[Y]$. Now, let $h, h' \in H$, and let $u \in N(h) \cap X$ and $v \in N(h') \cap X$. One can observe that a similar argument can be applied, by possibly starting the previous path with $(h, 1, u)$, in case $h = h_{ux}$ for some x not within the u, v-path P taken in $G[X]$.

Now, assume that Y is a closed connected set of $\mathcal{G}$, and consider $X = Y \cap V(G)$. We want to show that $G[X]$ has diameter at most 2. Suppose by contradiction that u and v are a distance at least 3 in $G[X]$. Observe that, since each $h \in H$ has degree exactly 2 in $\mathcal{G}$, we get that every temporal path between vertices of $V(G)$ in $\mathcal{G}$ is related to exactly one path of length at most 2 in G. More formally, if P is a temporal x, y-path in $\mathcal{G}$, with $x, y \in V(G)$, then P constrained to $V(G)$ is a x, y-path in G of length at most 2. Indeed, this occurs because at most two edges between vertices of H can be traversed. We then get a contradiction as $u, v \in Y$ and are at distance at least 3 in G.

Finally, we prove that every closed temporal unilateral connected set is also a closed temporal connected set. Since the reverse trivially holds, we get that it is also NP-complete to decide whether $Y \subseteq V(\mathcal{G})$ is a closed TUCC. So, consider $Y \subseteq V(\mathcal{G})$ a closed temporal unilateral connected set, and suppose that $x, y \in Y$ are such that x reaches y in $\mathcal{G}[Y]$. Let $(x = x_1, t_1, x_2, \ldots, x_q, t_q, x_{q+1} = y)$ be a temporal x, y-path in $\mathcal{G}[Y]$. We argue that $P = (y = x_{q+1}, 6 - t_q, x_q, \ldots, x_2, 6 - t_1, x_1 = x)$ is a temporal y, x-path in $\mathcal{G}$. For this, observe that if uv is an edge in $\mathcal{G}$ and $t \in \lambda(uv)$, then $6 - t \in \lambda(uv)$. Additionally, since $t_1 \leq t_2 \leq \ldots \leq t_q$, we have that $6 - t_q \leq 6 - t_{q-1} \leq \ldots \leq 6 - t_1$. It follows that indeed P is a temporal path, as desired. We then get that every closed temporal unilateral connected set is also a closed temporal connected set, as we wanted to prove. □

6. Concluding remarks

In this paper, we revisit the notion of connected components in temporal graphs introduced in [12] from the point of view of parameterized complexity. We then consider unilateral connectivity in temporal graphs, and investigate all related problems, in both the strict and the non-strict setting, as well as both for directed and undirected temporal graphs, parameterizing by the size k of the desired component, the lifetime τ of the considered (directed) temporal graph $\mathcal{G}$, and by $k + \tau$. We classify all possible entries in Table 1, leaving open just the following questions.

Question 4. *Given an undirected temporal graph $\mathcal{G}$, and considering parameterization by k, the size of the searched component, what is the complexity of deciding the existence of a closed TCC?*

Question 5. *Given a directed temporal graph $\mathcal{G}$ with lifetime 2, and considering parameterization by k, the size of the searched component, what is the complexity of deciding the existence of a closed TCC (TUCC)?*

We additionally prove a lower bound for testing connectivity, and prove that deciding maximality of closed (unilateral) connectivity is NP-complete.

CRediT authorship contribution statement

Isnard Lopes Costa: Writing – original draft, Writing – review & editing. **Raul Lopes:** Writing – original draft, Writing – review & editing. **Andrea Marino:** Writing – original draft, Writing – review & editing. **Ana Silva:** Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Raul Lopes reports financial support was provided by Paris Dauphine University - PSL. Isnard Lopes Costa, Ana Silva reports financial support was provided by Federal University of Ceara. Andrea Marino reports financial support was provided by University of Florence.

Data availability

No data was used for the research described in the article.

References

[1] P. Borgnat, E. Fleury, J. Guillaume, C. Magnien, C. Robardet, A. Scherrer, Evolving networks, in: Mining Massive Data Sets for Security, 2007, pp. 198–203.

[2] A. Casteigts, P. Flocchini, W. Quattrociocchi, N. Santoro, Time-varying graphs and dynamic networks, Int. J. Parallel Emerg. Distrib. Syst. 27 (2012) 387–408.
[3] M. Latapy, T. Viard, C. Magnien, Stream graphs and link streams for the modeling of interactions over time, Soc. Netw. Anal. Min. 8 (2018) 1–29.
[4] O. Michail, An introduction to temporal graphs: an algorithmic perspective, Internet Math. 12 (2016) 239–280.
[5] D. Kempe, J.M. Kleinberg, A. Kumar, Connectivity and inference problems for temporal networks, J. Comput. Syst. Sci. 64 (2002) 820–842.
[6] P. Zschoche, T. Fluschnik, H. Molter, R. Niedermeier, The complexity of finding small separators in temporal graphs, J. Comput. Syst. Sci. 107 (2020) 72–92.
[7] V. Nicosia, J. Tang, M. Musolesi, G. Russo, C. Mascolo, V. Latora, Components in time-varying graphs, Chaos, Interdiscip. J. Nonlinear Sci. 22 (2012) 023101.
[8] A. Casteigts, A.-S. Himmel, H. Molter, P. Zschoche, Finding temporal paths under waiting time constraints, Algorithmica 83 (2021) 2754–2802.
[9] J.A. Enright, K. Meeks, H. Molter, Counting temporal paths, in: P. Berenbrink, P. Bouyer, A. Dawar, M.M. Kanté (Eds.), 40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023, Hamburg, Germany, March 7–9, 2023, in: LIPIcs, vol. 254, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, 30.
[10] R. Haag, H. Molter, R. Niedermeier, M. Renken, Feedback edge sets in temporal graphs, Discrete Appl. Math. 307 (2022) 65–78.
[11] M. Rymar, H. Molter, A. Nichterlein, R. Niedermeier, Towards classifying the polynomial-time solvability of temporal betweenness centrality, J. Graph Algorithms Appl. 27 (2023) 173–194, https://doi.org/10.7155/jgaa.00619.
[12] S. Bhadra, A. Ferreira, Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks, in: Ad-Hoc, Mobile, and Wireless Networks, Second International Conference, ADHOC-NOW 2003 Montreal, Canada, October 8-10, 2003, Proceedings, 2003, pp. 259–270.
[13] E. Arjomandi, On finding all unilaterally connected components of a digraph, Inf. Process. Lett. 5 (1976) 8–10.
[14] A.B. Borodin, I. Munro, Notes on efficient and optimal algorithms, Technical Report, U. of Toronto and U. of Waterloo, Canada, 1972.
[15] R.G. Downey, M.R. Fellows, Fixed-parameter tractability and completeness II: on completeness for w [1], Theor. Comput. Sci. 141 (1995) 109–131.
[16] A. Casteigts, Finding structure in dynamic networks, arXiv preprint, arXiv:1807.07801, 2018.
[17] A. Casteigts, T. Corsini, W. Sarkar, Invited paper: simple, strict, proper, happy: a study of reachability in temporal graphs, in: S. Devismes, F. Petit, K. Altisen, G.A.D. Luna, A.F. Anta (Eds.), Stabilization, Safety, and Security of Distributed Systems - 24th International Symposium, SSS 2022, Clermont-Ferrand, France, November 15-17, 2022, Proceedings, in: Lecture Notes in Computer Science, vol. 13751, Springer, 2022, pp. 3–18.
[18] M. Calamai, P. Crescenzi, A. Marino, On computing the diameter of (weighted) link streams, ACM J. Exp. Algorithmics 27 (2022) 4.3:1–4.3:28.
[19] H. Wu, J. Cheng, S. Huang, Y. Ke, Y. Lu, Y. Xu, Path problems in temporal graphs, Proc. VLDB Endow. 7 (2014) 721–732.
[20] J. Hopcroft, R. Tarjan, Algorithm 447: efficient algorithms for graph manipulation, Commun. ACM 16 (1973) 372–378.
[21] E.L. Lawler, J.K. Lenstra, A. Rinnooy Kan, Generating all maximal independent sets: NP-hardness and polynomial-time algorithms, SIAM J. Comput. 9 (1980) 558–565.
[22] A. Conte, R. Grossi, A. Marino, L. Versari, Listing maximal subgraphs satisfying strongly accessible properties, SIAM J. Discrete Math. 33 (2019) 587–613.
[23] C. Bron, J. Kerbosch, Algorithm 457: finding all cliques of an undirected graph, Commun. ACM 16 (1973) 575–577.
[24] J.D. Eblen, C.A. Phillips, G.L. Rogers, M.A. Langston, The maximum clique enumeration problem: algorithms, applications, and implementations, in: BMC Bioinformatics, vol. 13, Springer, 2012, pp. 1–11.
[25] D. Avis, K. Fukuda, Reverse search for enumeration, Discrete Appl. Math. 65 (1996) 21–46.
[26] A. Conte, R. Grossi, M.M. Kanté, A. Marino, T. Uno, K. Wasa, Listing induced Steiner subgraphs as a compact way to discover Steiner trees in graphs, in: MFCS, in: LIPIcs, vol. 138, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, 73.
[27] M.M. Kanté, V. Limouzy, A. Mary, L. Nourine, T. Uno, A polynomial delay algorithm for enumerating minimal dominating sets in chordal graphs, in: International Workshop on Graph-Theoretic Concepts in Computer Science, Springer, 2015, pp. 138–153.
[28] S. Balev, Y. Pigné, E. Sanlaville, J. Schoeters, Temporally connected components, Technical Report hal-03966327, Normandie Univ, UNIHAVRE, LITIS, 2023.
[29] R. Impagliazzo, R. Paturi, On the complexity of k-SAT, J. Comput. Syst. Sci. 62 (2001) 367–375.
[30] A. Jarry, Z. Lotker, Connectivity in evolving graph with geometric properties, in: Proceedings of the 2004 Joint Workshop on Foundations of Mobile Computing, 2004, pp. 24–30.
[31] J. Moon, L. Moser, On cliques in graphs, Isr. J. Math. 3 (1965) 23–28, https://doi.org/10.1007/BF02760024.
[32] R. Becker, A. Casteigts, P. Crescenzi, B. Kodric, M. Renken, M. Raskin, V. Zamaraev, Giant components in random temporal graphs, in: N. Megow, A.D. Smith (Eds.), Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2023, September 11-13, 2023, Atlanta, Georgia, USA, in: LIPIcs, vol. 275, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, 29.
[33] A. Casteigts, M. Raskin, M. Renken, V. Zamaraev, Sharp thresholds in random simple temporal graphs, in: FOCS, IEEE, 2021, pp. 319–326.
[34] V. Nicosia, J. Tang, C. Mascolo, M. Musolesi, G. Russo, V. Latora, Graph metrics for temporal networks, in: Temporal Networks, Springer, 2013, pp. 15–40.
[35] C. Gómez-Calzado, A. Casteigts, A. Lafuente, M. Larrea, A connectivity model for agreement in dynamic systems, in: European Conference on Parallel Processing, Springer, 2015, pp. 333–345.
[36] M. Vernet, Y. Pigné, E. Sanlaville, A study of connectivity on dynamic graphs: computing persistent connected components, 4OR 21 (2023) 205–233, https://doi.org/10.1007/s10288-022-00507-3.
[37] E.C. Akrida, P.G. Spirakis, On verifying and maintaining connectivity of interval temporal networks, Parallel Process. Lett. 29 (2019) 1950009.
[38] V. Levorato, C. Petermann, Detection of communities in directed networks based on strongly p-connected components, in: 2011 International Conference on Computational Aspects of Social Networks (CASoN), IEEE, 2011, pp. 211–216.
[39] R. Sedgewick, K. Wayne, Algorithms: Part I, Addison-Wesley Professional, 2014.
[40] G.A. Cheston, A correction to a unilaterally connected components algorithm, Inf. Process. Lett. 7 (1978) 125, https://doi.org/10.1016/0020-0190(78)90058-3.
[41] J. Bang-Jensen, G.Z. Gutin, Digraphs: Theory, Algorithms and Applications, Springer Science & Business Media, 2008.
[42] T. Mchedlidze, A. Symvonis, Unilateral orientation of mixed graphs, in: International Conference on Current Trends in Theory and Practice of Computer Science, Springer, 2010, pp. 588–599.
[43] J.F. Fink, L. Lesniak-Foster, Graphs for which every unilateral orientation is traceable, Ars Comb. 9 (1980) 113–118.
[44] D.B. West, et al., Introduction to Graph Theory, vol. 2, Prentice Hall, Upper Saddle River, 2001.
[45] M. Cygan, F.V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, Springer, 2015.
[46] R.G. Downey, M.R. Fellows, Fundamentals of Parameterized Complexity, Texts in Computer Science, Springer, 2013.
[47] R. Peeters, The maximum edge biclique problem is np-complete, Discrete Appl. Math. 131 (2003) 651–654, https://doi.org/10.1016/S0166-218X(03)00333-0, https://www.sciencedirect.com/science/article/pii/S0166218X03003330.
[48] M. Yannakakis, Computing the minimum fill-in is np-complete, SIAM J. Algebraic Discrete Methods 2 (1981) 77–79.
[49] M. Borassi, P. Crescenzi, M. Habib, Into the square: on the complexity of some quadratic-time solvable problems, in: ICTCS, in: Electronic Notes in Theoretical Computer Science, vol. 322, Elsevier, 2015, pp. 51–67.
[50] M. Pătraşcu, R. Williams, On the possibility of faster SAT algorithms, in: Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2010, pp. 1065–1075.

[51] V.V. Williams, R. Williams, Subcubic equivalences between path, matrix and triangle problems, in: 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, IEEE, 2010, pp. 645–654.
[52] F.M. Pajouh, B. Balasundaram, On inclusionwise maximal and maximum cardinality k-clubs in graphs, Discrete Optim. 9 (2012) 84–97.